Chapter 11
DESIGN, PROTOTYPING and CONSTRUCTION
The Wooting One
Overview

• Prototyping
• Conceptual design
• Concrete design
• Using scenarios
• Generating prototypes
• Construction

• Uitdagingen uit de praktijk & tentamentraining.
Prototyping

• What is a prototype?
• Why prototype?
• Different kinds of prototyping
  - Low fidelity
  - High fidelity
• Compromises in prototyping
  - Vertical
  - Horizontal
• Final product needs to be engineered
What is a prototype?

In other design fields a prototype is a small-scale model:

- a miniature car
- a miniature building or town
- the examples here come from a 3D printer

Figure 11.1 (a) Color output from a 3D printer: all the gears and rods in this model were ‘printed’ in one pass from bottom to top, and when one gear is turned, the others turn too.

Source: (a) The Computer Language Company, Inc., courtesy of Alan Freedman
(c) A teddy bear ‘printed’ from a wireframe design
http://www.disneyresearch.com/project/printed-teddy-bears/
(c) Courtesy of Scott Hudson, Human–Computer Interaction Institute, Carnegie Mellon University.
What is a prototype?

In interaction design it can be (among other things):

• a series of screen sketches
• a storyboard, i.e. a cartoon-like series of scenes
• a Powerpoint slide show
• a video simulating the use of a system
• a lump of wood (e.g. PalmPilot)
• a cardboard mock-up
• a piece of software with limited functionality written in the target language or in another language
Why prototype?

• Evaluation and feedback are central to interaction design

• Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing

• Team members can communicate effectively

• You can test out ideas for yourself

• It encourages reflection: very important aspect of design

• Prototypes answer questions, and support designers in choosing between alternatives
## (Filtering) dimensions of prototyping

<table>
<thead>
<tr>
<th>Filtering dimension</th>
<th>Example variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance</td>
<td>size; color; shape; margin; form; weight; texture; proportion; hardness; transparency; gradation; haptic; sound</td>
</tr>
<tr>
<td>Data</td>
<td>data size; data type (e.g., number; string; media); data use; privacy type; hierarchy; organization</td>
</tr>
<tr>
<td>Functionality</td>
<td>system function; users’ functionality need</td>
</tr>
<tr>
<td>Interactivity</td>
<td>input behavior; output behavior; feedback behavior; information behavior</td>
</tr>
<tr>
<td>Spatial structure</td>
<td>arrangement of interface or information elements; relationship among interface or information elements – which can be either two-or three-dimensional, intangible or tangible, or mixed</td>
</tr>
</tbody>
</table>
### Manifestation dimensions of prototyping

<table>
<thead>
<tr>
<th>Manifestation dimension</th>
<th>Definition</th>
<th>Example variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>Medium (either visible or invisible) used to form a prototype</td>
<td>Physical media, e.g. paper, wood, and plastic; tools for manipulating physical matters, e.g. knife, scissors, pen, and sandpaper; computational prototyping tools, e.g. Macromedia Flash and Visual Basic; physical computing tools, e.g. Phidgets and Basic Stamps; available existing artifacts, e.g. a beeper to simulate a heart attack</td>
</tr>
<tr>
<td>Resolution</td>
<td>Level of detail or sophistication of what is manifested (corresponding to fidelity)</td>
<td>Accuracy of performance, e.g. feedback time responding to an input by a user (giving user feedback in a paper prototype is slower than in a computer-based one); appearance details; interactivity details; realistic versus faked data</td>
</tr>
<tr>
<td>Scope</td>
<td>Range of what is covered to be manifested</td>
<td>Level of contextualization, e.g. website color scheme testing with only color scheme charts or color schemes placed in a website layout structure; book search navigation usability testing with only the book search related interface or the whole navigation interface</td>
</tr>
</tbody>
</table>

Table 11.2 The definition and variables of each manifestation dimension
What to prototype?

- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas
Low-fidelity Prototyping

• Uses a medium which is unlike the final medium, e.g. paper, cardboard

• Is quick, cheap and easily changed

• Examples:
  – sketches of screens, task sequences, etc
  – ‘post-it’ notes
  – storyboards
  – ‘Wizard-of-Oz’
Storyboards

• Often used with scenarios, bringing more detail, and a chance to role play

• It is a series of sketches showing how a user might progress through a task using the device

• Used early in design
## Example storyboard

<table>
<thead>
<tr>
<th>Type</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Low-fidelity prototype| - Lower development cost  
- Evaluates multiple design concepts  
- Useful communication device  
- Addresses screen layout issues  
- Useful for identifying market requirements  
- Proof of concept | - Limited error checking  
- Poor detailed specification to code to Facilitator-driven  
- Limited utility after requirements established  
- Limited usefulness for usability tests  
- Navigational and flow limitations |
| High-fidelity prototype| - Complete functionality  
- Fully interactive  
- User-driven  
- Clearly defines navigational scheme  
- Use for exploration and test  
- Look and feel of final product  
- Serves as a living specification  
- Marketing and sales tool | - More resource-intensive to develop  
- Time-consuming to create  
- Inefficient for proof-of-concept designs  
- Not effective for requirements gathering |
Sketching

- Sketching is important to low-fidelity prototyping
- Don’t be inhibited about drawing ability. Practice simple symbols

Figure 11.5 A storyboard depicting how to fill a car with gas
Card-based prototypes

- Index cards (3 X 5 inches)
- Each card represents one screen or part of screen
- Often used in app and website development
‘Wizard-of-Oz’ as lofi prototyping

• The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.

• Usually done early in design to understand users’ expectations
High-fidelity prototyping

• Uses materials that you would expect to be in the final product
• Prototype looks more like the final system than a low-fidelity version
• High-fidelity prototypes can be developed by integrating existing hardware and software components
• Danger that users think they have a complete system……..see compromises
‘Wizard-of-Oz’ as hifi prototyping

• The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.

• Usually done early in design to understand users’ expectations
Compromises in prototyping

• All prototypes involve compromises

• For software-based prototyping maybe there is a slow response? sketchy icons? limited functionality?

• Two common types of compromise
  • horizontal: provide a wide range of functions, but with little detail
  • vertical: provide a lot of detail for only a few functions

• Compromises in prototypes mustn't be ignored. Product needs engineering
Conceptual design

• Transform user requirements/needs into a conceptual model

• A conceptual model is an outline of what people can do with a product and what concepts are needed to understand and interact with it

• Mood board may be used to capture feelings

• Consider alternatives: prototyping helps
Is there a suitable metaphor?

• Interface metaphors combine familiar knowledge with new knowledge in a way that will help the user understand the product.

• Three steps: understand functionality, identify potential problem areas, generate metaphors

• Evaluate metaphors:
  
  How much structure does it provide?
  How much is relevant to the problem?
  Is it easy to represent?
  Will the audience understand it?
  How extensible is it?
Considering interaction and interface types

• Which interaction type?
  – How the user invokes actions
  – Instructing, conversing, manipulating or exploring

• Do different interface types provide insight?
  – shareable, tangible, augmented reality, etc.
Expanding the initial conceptual model

• What functions will the product perform?
  - What will the product do and what will the human do (task allocation)?

• How are the functions related to each other?
  - Sequential or parallel?
  - Categorisations, e.g. all actions related to privacy on a smartphone

• What information is needed?
  - What data is required to perform the task?
  - How is this data to be transformed by the system?
Concrete design

• Many aspects to concrete design
  – Color, icons, buttons, interaction devices etc.

• User characteristics and context
  – Accessibility, cross-cultural design

• Cultural website guidelines

  successful products “are … bundles of social solutions. Inventors succeed in a particular culture because they understand the values, institutional arrangements, and economic notions of that culture.”
Using scenarios

- Express proposed or imagined situations

- Used throughout design in various ways
  - as a basis for overall design
  - scripts for user evaluation of prototypes
  - concrete examples of tasks
  - as a means of co-operation across professional boundaries

- Plus and minus scenarios to explore extreme cases
Generate storyboard from scenario

Figure 11.4 Some simple sketches for low-fidelity prototyping
Generate card-based prototype from use case

Figure 11.6 Prototype developed for cell phone user interface
Explore the user’s experience

• Use personas, card-based prototypes or stickies to model the user experience

• Visual representation called:
  – design map
  – customer/user journey map
  – experience map

• Two common representations
  – wheel
  – timeline
An experience map drawn as a wheel

Figure 11.19 (a) An experience map using a wheel representation. (b) An example timeline design map illustrating how to capture different issues.

An experience map drawn as a timeline

Design Map: Megan delivers the presentation
(last updated by TA, 2/18/06)

- Megan logs on to the presentation system
- Megan sees the presenter page
- Megan sees that her slides are ready and she does a last-minute flip through
- Megan fixes an error on one of her slides
- Megan sees that audience members are starting to arrive

Should we let Megan log on if Ivan hasn't set everything up yet?

The presenter page should reassure her that the streams are started and the preso is good to go.

Let's create a way for her to flip through her slides (and change them?) without any audience members seeing this process.

Are we going to be able to support this kind of last-minute change?

Megan has already uploaded all of her slides.

The presenter page should let her see that Ivan is there already.

What if an audience member tries to connect before Megan or even before Ivan?

See Design Map: Sam arrives for the Presentation.

See a related Design Map

(b)

Figure 11.19 Continued
Construction: physical computing

• Build and code prototypes using electronics

• Toolkits available include
  – Arduino
  – LilyPad (for fabrics)
  – Senseboard
  – MaKey MaKey

• Designed for use by wide range of people
Physical computing kits

Figure 11.22 The Arduino board

Source: Courtesy of Nicolai Marquardt
Physical computing kits

Figure 11.24 The MaKey MaKey toolkit
Physical computing kits

Figure 11.25 A group of retired friends playing with a MaKey MaKey toolkit
What is BITalino? Is it for you?

DIY toolkit turns people's bodies into data-tracking machines

Board Kit
what's included
The one and only standard toolkit with our awesome all-in-one board.

Freestyle Kit
what's included
We break out all the blocks for you to make custom designs fast & easy.

Plugged Kit
what's included
Cabled sensor connection for interchangeable use of our sensors and custom designs as well.

OpenSignals
show me more
Our signature software for real-time data visualization and recording.
Doing cyber-UX (1)
Construction: SDKs

• Software Development Kits
  – programming tools and components to develop for a specific platform, e.g. iOS

• Includes: IDE (Integrated Development Environment), documentation, drivers, sample code, application programming interfaces (APIs)

• Makes development much easier

• Microsoft’s Kinect SDK has been used in research
Summary

• Different kinds of prototyping are used for different purposes and at different stages

• Prototypes answer questions

• The final product must be engineered appropriately

• Two aspects of design: conceptual and concrete

• To generate conceptual design, consider interface metaphors, interaction types and interface types

• Storyboards can be generated from scenarios

• Card-based prototypes can be generated from use cases

• Physical computing kits and SDKs facilitate transition from design to construction
Chapter 12
INTERACTION DESIGN IN PRACTICE
Overview

- Recap / the old stuff
- Agile UX
- Lean UX
- Design Patterns
- Customer journey mapping
recap
Waterfall model

1. Requirements Analysis
2. Design
3. Development
4. Testing
5. Maintenance

Software Development Life Cycle
designing the user experience
Agile development

• Short (one to three week) timeboxes of iterative development (sprint, iteration, cycle)

• Early and repeated customer/user feedback

• Re-prioritisation of work based on customer/user so that emergent requirements can be handled

• Many approaches, e.g. eXtreme Programming (XP), Scrum, DSDM
AgileUX

• Integrates techniques from interaction design and Agile software development

• AgileUX requires a change of mindset

• In Agile, as implementation proceeds:
  – requirements are elaborated
  – requirements are re-prioritised

• All techniques in UX are still relevant but when and how much needs re-thinking
  – focus on product, not design, as deliverable
  – cross-functional teams

• Three practical areas: user research, aligning work practices, documentation
User research

• Aims to characterise users through data collection and analysis

• Agile’s timeboxing approach does not support long periods of user research

• User evaluations and some detailed work can be fitted within a timebox

• Some user research can be performed in iteration 0 (zero), before implementation starts

• Ongoing programme of user research
Agile Usability Testing & User Experience Analytics

01 LISTEN
02 RESEARCH
03 TEST & ACT
04 MEASURE
Old versus new style (1)

Waterfall UX (Bad!)

Agile UX (Good!)

Project time
Old versus new style (2)

**Figure 1.** In a waterfall development cycle, analysis, design, coding, and quality assurance testing are separate stages of a software release that spans months or years. In Agile development, each of a set of incremental mini-releases (each created in 2-4 weeks) has these stages. Adapted from *Cutter Consortium* [8].
Old versus new style (4): Theory versus practice

Figure 2. In the perfect theoretical version of waterfall development, usability investigations contributing to the analysis and design phases were supposed to precede coding, but in reality developers would begin coding immediately.
### Old versus new style (3)

<table>
<thead>
<tr>
<th>Heavy-Weight Processes</th>
<th>Light-Weight Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed, up-to-date documentations and models</td>
<td>Cards and hand-drawn abstract models. Travel light. Communicate rather than document.</td>
</tr>
<tr>
<td>High-fidelity prototypes</td>
<td>Abstract prototypes, use simplest tools</td>
</tr>
<tr>
<td>Develop and prove concepts with user feedback. Iterate.</td>
<td>Courage. Design for needs (user's tasks) rather than user expectations. Retrieve design from models rather than continuous user feedback.</td>
</tr>
<tr>
<td>Time-consuming usability evaluations, workshops with intense stakeholder integration</td>
<td>Fast usability inspections. No need to evaluate if models are right.</td>
</tr>
<tr>
<td>Problem</td>
<td>Symptoms</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Not enough design time</td>
<td>Developers waiting on designs</td>
</tr>
<tr>
<td>Sprints are too short</td>
<td>Designs can’t be finished in time</td>
</tr>
<tr>
<td></td>
<td>No time for usability testing</td>
</tr>
<tr>
<td></td>
<td>No time to set up customer contact</td>
</tr>
<tr>
<td></td>
<td>Feedback not early enough</td>
</tr>
<tr>
<td></td>
<td>No data to act on – opinions rule</td>
</tr>
<tr>
<td></td>
<td>Product isn’t validated</td>
</tr>
<tr>
<td></td>
<td>End-users and clients won’t participate</td>
</tr>
<tr>
<td></td>
<td>Can’t get buy-in from rest of team</td>
</tr>
<tr>
<td></td>
<td>Non informed decisions are made</td>
</tr>
<tr>
<td>Weak Agile &quot;customer&quot; [6]</td>
<td>UX time spent in many meetings instead of on designs and iterations</td>
</tr>
<tr>
<td></td>
<td>Demoralized by UX quality drop</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>UX is not full-time on one Agile team</td>
<td>Large backlog of features/bugs</td>
</tr>
<tr>
<td></td>
<td>Prioritization feedback ignored</td>
</tr>
<tr>
<td></td>
<td>No control over timing of designs</td>
</tr>
<tr>
<td>No sprint/cycle planning</td>
<td>Feature set is cast in stone</td>
</tr>
<tr>
<td></td>
<td>No time to incorporate changes</td>
</tr>
<tr>
<td></td>
<td>No re-ordering of features is allowed</td>
</tr>
<tr>
<td>User feedback is ignored</td>
<td>Missing the &quot;big picture&quot;</td>
</tr>
<tr>
<td></td>
<td>No shared vision or end goal</td>
</tr>
<tr>
<td></td>
<td>Too much focus on details</td>
</tr>
<tr>
<td></td>
<td>Hard to prioritize/make design decisions</td>
</tr>
<tr>
<td>Poor communication</td>
<td>Misunderstood designs</td>
</tr>
<tr>
<td></td>
<td>Agile team doesn’t buy-into designs</td>
</tr>
<tr>
<td></td>
<td>Important information is lost</td>
</tr>
<tr>
<td>Team isn't co-located</td>
<td>No sense of team - lack of trust</td>
</tr>
<tr>
<td></td>
<td>Language and/or time barriers</td>
</tr>
<tr>
<td></td>
<td>Not enough communication</td>
</tr>
<tr>
<td>Dependency issues</td>
<td>Requiring input from non-Agile teams (e.g., marketing sign-offs, lawyers)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Aligning work practices

• Designing a complete product upfront causes problems because of re-prioritisation

• Some upfront work is needed (technical and UX)

• Use a parallel tracks approach:
  – create product vision before development starts
  – do design work one iteration ahead of development
  – some teams work two iterations ahead
Parallel tracks approach to AgileUX

Figure 12.2 Cycle 0 and its relationship to later cycles
Aligning work practices

• Advantages of parallel tracks approach:
  – no design time wasted on features not implemented
  – usability testing and contextual inquiry could be done on the same customer visit, saving time
  – timely feedback on the designs was received from developers and customers
  – Agile flexibility supports schedule changes if a problem is found

• Parallel tracks is commonly used
Lean UX cycle

THINK
- Generative Research
- Ideation
- Mental models
- Behavior Models
- Test Results
- Competitive Analysis

CHECK
- A/B Testing
- Site Analytics
- Usability Testing
- Funnel
- Sign-ups

MAKE
- Prototypes
- Wireframes
- Value Prop
- Landing Page
- Hypotheses
- Comps
- Deployed Code

Reduce cycle time, not build time
Typical questions

- Who are our users?
- What is the product used for?
- When is it used?
- What situations is it used in?
- What will be the most important functionality?
- What’s the biggest risk to product delivery?
Minimum Viable Product (MVP)

1. Declare Assumptions / Initial Understanding
2. Create an MVP
3. Run an Experiment
4. Feedback and Research

How not to build a minimum viable product:
1. 2. 3. 4.

How to build a minimum viable product:
1. 2. 3. 4. 5.

Kirill Shikhanov “MVP”. https://dribbble.com/shots/1753131-MVP
### A Lean UX Design Process

#### Annie Wang

**I'm a UX Team of One.**
- Wire Framing
- Interaction Design
- Visual Design & Style Guide
  - Conceptual Model
  - Information Architecture
  - User Research
  - Discovery

**Design Can Change Business Design Thinking**

**User Experience**
- Business
- Product
- Technology

**UX as a Facilitator, Not Just a Designer**

**Lean = Skinny Reduce Inventory**
- Think
  - Generative Research
  - Ideation
- Mental Models
- Behavior Models
- Test Results
- Competitive Analysis

**Make**
- Prototypes
- Wireframes
- Value Prop
- Landing Page
- Hypothesis
- Corps
- Deployed Code

**Check**
- AB Testing
- Site Analytics
- Usability Testing
- Funnel
- Sign-up

---

#### My Lean UX Process

**What & How for Startups I Worked**

**Discovery**
- Discover and research to define user scenario and conceptual model

**Wireframe**
- Use Cases
- Work Flow
- Site Map
- Screen Wireframing
- Major Interaction

**Prototype**
- Visual Design
- Icons + Style Guide
- Screen Mockup
- HTML Prototype
- Flash Prototype
- IOS Prototype

**Validate Internally**
- Get quick and frequent feedback from executive and team members

**Test Externally**
- Meet with customers, find niche group of target users

**Summarize**
- Validate and learn from user behavior

**Iterate**
- Modify
- Update
- Plan and move to next cycle

---

#### User Feedback is the Key

**@Jeff Gothelf**

**Ideal vs. Realistic vs. Solution**

**Ideal**
- Objectives

**Realistic**
- Objectives

**Solution**
- Blank Page

---

#### My Design Philosophy

**Clean & Simple**

**Less is More**

---

**www.anniestudio.org**

annie@anniestudio.org
THE DIFFERENCE BETWEEN Lean UX and Agile UX

**Lean UX**
Application of User Experience Design methods into product development, tailored to fit Build-Measure-Learn cycles.

Design united with business and development.

Methods
Inspired by startup development and The Lean Startup Methodology.

**Agile UX**
An attempt to integrate User Experience Design and Agile Software Development Methodology.

UX Design team & Dev team working together.

Methodology
Inspired by Agile Manifesto. Forced to blossom by growing popularity of Agile Software Development Methodology.

Try Lean UX & Agile UX in UXPin - The UX Design App (http://uxpin.com)
Documentation

• Most common communication approach for UX designers

• Agile discourages this kind of communication, in favour of discussion

• Only use documentation where needed. Ask:
  – Who will read it?
  – Who will use it?
  – What is the minimum needed?
  – Is there duplication anywhere?
  – How polished does it need to be?
Figure 12.3 (a) A low-fidelity user journey

Documentation: how polished?

(b) A high-fidelity user journey

Design Patterns

• Capture design experience:
  – a solution to a problem in a context
  – can be instantiated in many ways: generative

• Patterns may be individual, in languages, in catalogues, or in software components, e.g. Github or platform websites

• Carousel pattern as an example:
Resources: Mobile UI Design Patterns
Design Patterns: Carousels (1)

• Create a Fanciful Amusement, Not a House of Horrors

• Most ubiquitous type: the feature area carousel

• Positives
  – more than one piece of content to occupy the same piece of prime real estate on the homepage
  – appears near the top of the viewable area

• Negatives
  – people often immediately scroll past these large images
  – get the wrong idea about an organization
Design Patterns:
Carousels (2), guidelines

1. <= 5 frames
2. Use crisp-looking text and images that coincide with the organization's charter.
3. Indicate how many frames are present, and where the user is within the “progression.” This helps people feel in control.
4. Use icons and links that are understandable and recognizable.
5. Ensure that navigation controls appear inside the carousel.
6. If offering a navigation button for each frame (rather than arrows to scroll through), ensure that each button looks different, and represents the frame.
7. Make links and buttons large enough to decipher and click.
<table>
<thead>
<tr>
<th>User-Centered Design</th>
<th>Usage-Centered Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Focus</strong> is on users: user experience and user satisfaction</td>
<td><strong>Focus</strong> is on usage: improved tools supporting task accomplishment</td>
</tr>
<tr>
<td>Driven by user input</td>
<td>Driven by models and modeling</td>
</tr>
<tr>
<td>Substantial user involvement</td>
<td>Selective user involvement</td>
</tr>
<tr>
<td>• User studies</td>
<td>• Explorative modeling</td>
</tr>
<tr>
<td>• Participatory design</td>
<td>• Model variation</td>
</tr>
<tr>
<td>• User feedback</td>
<td>• Usability inspections</td>
</tr>
<tr>
<td>• User testing</td>
<td>Design by modeling</td>
</tr>
<tr>
<td>Design by iterative prototyping</td>
<td>Systematic, fully specified process</td>
</tr>
<tr>
<td>Highly varied, informal, or unspecified processes</td>
<td>Design by engineering</td>
</tr>
<tr>
<td>Design by trial-and-error, evolution</td>
<td></td>
</tr>
</tbody>
</table>
Tools for Interaction Design

• Tools support all aspects of the design process:
  – creativity, sketching, simulation, brainstorming, library search, mindmapping, video capture

• Tools integrate together to speed up prototyping

• Interactive wireframes or mockups can be produced using, e.g.
  – Balsamiq©
  – Axure©

• Higher fidelity prototype can be produced by linking interactive wireframe to design pattern library with software components
Customer journey mapping

[ DIGITAL TOUCHPOINTS ]

- EMAIL
- WEB SITES/LANDING PAGES
- SOCIAL MEDIA
- 3RD PARTY SITES
- MOBILE APP/SITE
- COMMUNITY
- CHAT
- TWITTER/SOCIAL
- WEB SELF-SERVICE

[ PHYSICAL TOUCHPOINTS ]

- WORD-OF-MOUTH
- DIRECT MAIL
- STORE/BRANCH
- AGENT/BROKER
- CALL CENTER/IVR
- OFFERS TO CUSTOMERS
- EMAIL
- LOYALTY PROGRAM
- SURVEY
- MAILINGS
- OFFERS IN INVOICE
The future of travel is likely to be shaped by technological and social innovations to reduce stress, uncertainty and to encourage collaboration among travellers and with travel providers. Click below to share the frustrations and desires you agree with.

<table>
<thead>
<tr>
<th>Travellers’ Frustrations</th>
<th>Travellers’ Expectations</th>
</tr>
</thead>
<tbody>
<tr>
<td>At the airport - automatic transit 47% going through security, passport control and customs</td>
<td>Automated identity management systems to track flows of people going through security and border control 53% happy: to provide more personal information for efficient travel</td>
</tr>
<tr>
<td>Taking the stress out of travel 51% over 50s - going through customs or passport control</td>
<td>Intelligent tickets can update the traveller on delays or changes 82% desire: a device to monitor and reduce levels of stress while travelling</td>
</tr>
<tr>
<td>Finding your way 47% time to research options</td>
<td>Intelligent recommendations information from expert blogs and online guides help create personalised travel guides 86% desire: a personalised travel guide which aggregates recommendations</td>
</tr>
<tr>
<td>Next generation of experience 51% not knowing where the best local restaurants, bars and venues are</td>
<td>Augmented reality access historical and cultural information about a place and access peer reviews 61% desire: a mobile application that overlays visual information about the physical world</td>
</tr>
<tr>
<td>Travel services 47% lack of good quality advice</td>
<td>Payment with memory memory of expenditure will enable intelligent passenger records 47% happy: to provide personal data for a more personalised service</td>
</tr>
<tr>
<td>Work life balance for business tourists 51% not having access to home music, video and data</td>
<td>Cloud computing easy access to your own music, video and data from hotel rooms 59% business travellers desire: complete access to everything they can get in their home</td>
</tr>
</tbody>
</table>
Overview of Customer-Centered Design

“You’ve gotta start with the customer experience and work back to the technology”
- Steve Jobs. WWDC 1997

Customer-Centered Design Process

1. Observe and empathize with customers' needs
2. Collect data
3. Brainstorm
4. Design a prototype based on observations
5. Test
6. Refine and make adjustments
7. Develop

Customer-Centered Design Checklist

- Ethnographic observations
- Concept ideation
Customer-Centered Design Checklist
- Ethnographic observations
- Customer interviews
- Analysis of customer findings
- Concept ideation
- Rapid prototyping
- Concept validation by the customer

CDD Benefits
- Products that solve real customer problems
- Products that customers want to use
- Products that customers want to talk about!

Technology driven
- Component focused
- System driven use cases
- Success based on functional features
- System performance is a premium

Vs

CDD driven
- Customer needs focused
- Real-world use cases
- Success based on customer values
- Customer engagement is a premium

How can Customer-Centered Design

LEAN UX
Lean UX can use customer centered design throughout all phases of the project lifecycle to validate product value, market fit and the customer experience.
How can Customer-Centered Design fit into Agile, Waterfall and Lean UX?

WATERFALL
Waterfall’s linear approach makes the beginning of the product design life cycle the best place for customer centered design to have an impact. Using it at the end of the process is possible but can be costly.

Customer-Centered Design in a Waterfall process

Lean UX
Lean UX can use customer centered design throughout all phases of the project lifecycle to validate product value, market fit and the customer experience.

Customer-Centered Design in a Lean UX process

When to Consider
- Limited design time
- Solution based iterations
- Team

42% of online projects are underfunded and under estimated.
Customer-Centered Design in a Waterfall process

- Requirement
- Design
- Development
- Testing & Validation
- Deployment & Maintenance

When to Consider
- Limited design time
- Solution based iterations
- Team collaboration

42% of online projects are underfunded and under manned when it comes to UX

When to Consider
- Fixed-scope
- Fixed-price
- Clear vision

15% of fail points can be fixed with the addition of customer centered design into a waterfall process

AGILE
Agile allows customer centered design to have an impact within each sprint but more at a component or feature level. Concept pivots can be challenging.

Customer-Centered Design in an Agile Process

The cost of not using Customer-Centered Design
The cost of not using Customer-Centered Design

$83 Billion in lost annually due to poor customer experiences online (US)

100x more expensive to correct a problem after release vs solving in the design phase

Learn more about Customer-Centered Design:
- Contextual Design: Defining Customer-Centered Systems -(by Hugh Beyer and Karen Holtzblatt)
- Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation (by Tim Brown)
- Design of Sites, The: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience (by By Douglas K. van Duyne, James A. Landay, Jason I. Hong)

References:
Summary

- AgileUX refers to approaches that integrate UX design and agile development
  - it requires a change in mindset by designers and developers
  - requirements are repeatedly re-prioritised, which aims to avoid wasted effort
  - UX design activities need rethinking: when, how much, and how to take forward
- Design patterns present a solution to a problem in a context
- Open source resources, e.g. on Github, make development of standard applications easier and quicker
- A range of automated tools to support interaction design in practice is available