Chapter 10:
ESTABLISHING REQUIREMENTS

Chapter 11:
DESIGN, PROTOTYPING AND CONSTRUCTION

Chapter 12:
INTERACTION DESIGN IN PRACTICE
ESTABLISHING REQUIREMENTS

Chapter 10
Overview

- The importance of requirements
- Different types of requirements
- Data gathering for requirements
- Data analysis and presentation
- Task description: Scenarios
  - Use Cases
  - Essential use cases
- Task analysis: HTA
What, how and why?

What needs to be achieved?

1. Understand as much as possible about users, task, context

2. Produce a stable set of requirements

How can this be done? (all of this is iterative)

- Data gathering activities
- Data analysis activities
- Presentation
- Express findings as requirements
What, how and why?

Why bother?

- Requirements definition is the stage where failure occurs most commonly
- Getting requirements right is crucial
What, how and why?
De behoefte aan validatie: een verkennend onderzoek naar requirements traceability in Agile Development

Bachelorthesis Informatiekunde
Requirements engineering

Course code: INFOMRE
Credits: 7.5 ECTS
Period: period 4 (week 17 through 26, i.e., 23-4-2018 through 29-6-2018; retake week 28)
Timeslot: C
Participants: up till now 67 subscriptions
Schedule: Official schedule representation can be found in Osiris

<table>
<thead>
<tr>
<th>form</th>
<th>group</th>
<th>time</th>
<th>week</th>
<th>room</th>
<th>teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecture</td>
<td></td>
<td>Mon 13.15-17.00</td>
<td>17-20</td>
<td>BBG-161</td>
<td>Fabiano Dalpiaz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mon 22-25</td>
<td></td>
<td>BBG-161</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thu 9.00-10.45</td>
<td>17-18</td>
<td>BBG-201</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thu 20-25</td>
<td></td>
<td>BBG-201</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thu 11.00-12.45</td>
<td>20</td>
<td>BBG-065</td>
<td></td>
</tr>
</tbody>
</table>
The Requirements Engineering Lab at Utrecht University!

Fabiano Dalpiaz
Sjaak Brinkkemper
Marcela Ruiz
F. Basak Aydemir
Sietse Overbeek
Gerard Wagenaar
Davide Dell’Anna
Natural Language Processing applied to RE

User stories (txt, csv) → Visual Narrator → Conceptual model

Visualization tools
- Interactive Narrator
- REVV
- Your project?

@2018 Fabiano Dalpiaz
Establishing requirements

- What do users want? What do users ‘need’?

  Requirements need clarification, refinement, completion, re-scoping

  Input: Requirements document (maybe)
  Output: stable requirements

- Why ‘establish’?

  Requirements arise from understanding users’ needs
  Requirements can be justified
Different kinds of requirements

Environment or context of use:

• physical: dusty? noisy? vibration? light? heat? humidity? …. (e.g. ATM)

• social: sharing of files, of displays, in paper, across great distances, synchronous, privacy for clients

• organisational: hierarchy, IT department’s attitude and remit, user support, communications structure and infrastructure, availability of training
Different kinds of requirements

• Functional:
  — what the system should do

• Non-Functional:
  — e.g. security, response time, speed, cost
Prioritizing requirements: MoSCoW (niet in DIT boek)

**Must have**
- critical, if even one Must have requirement is not included, the project delivery should be considered a failure

**Should have**
- important but not necessary

**Could have**
- desirable but not necessary, and could improve user experience or customer

**Won't have (this time)**
- least-critical, or not appropriate at the time
Prioritizing requirements: MoSCoW (niet in DIT boek)
Volere shell

Requirement #: 75

Description: The product shall issue an alert if a weather station fails to transmit readings.

Rationale: Failure to transmit readings might indicate that the weather station is faulty and needs maintenance, and that the data used to predict freezing roads may be incomplete.

Source: Road Engineers
Fit Criterion: For each weather station the product shall communicate to the user when the recorded number of each type of reading per hour is not within the manufacturer’s specified range of the expected number of readings per hour.

Customer Satisfaction: 3  Customer Dissatisfaction: 5
Dependencies: None  Conflicts: None
Supporting Materials: Specification of Rosa Weather Station
History: Raised by GBS, 28 July 99
# Volere requirements template

## PROJECT DRIVERS
1. The Purpose of the Product
2. The Stakeholders

## PROJECT CONSTRAINTS
3. Mandated Constraints
4. Naming Conventions and Definitions
5. Relevant Facts and Assumptions

## FUNCTIONAL REQUIREMENTS
6. The Scope of the Work
7. Business Data Model and Data Dictionary
8. The Scope of the Product
9. Functional and Data Requirements

## NON-FUNCTIONAL REQUIREMENTS
10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

## PROJECT ISSUES
18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions
Users: Who are they?

- Characteristics: nationality, educational background, attitude to computers
- System use: novice, expert, casual, frequent
  - Novice: prompted, constrained, clear
  - Expert: flexibility, access/power
  - Frequent: short cuts
  - Casual/infrequent: clear menu paths
What are the users’ capabilities?

Humans vary in many dimensions:

— size of hands may affect the size and positioning of input buttons

— motor abilities may affect the suitability of certain input and output devices

— height if designing a physical kiosk

— strength - a child’s toy requires little strength to operate, but greater strength to change batteries

— disabilities (e.g. sight, hearing)
**Personas**

- Capture a set of user characteristics (user profile)
- Not real people, but synthesised from real users
- Should not be idealised
- Bring them to life with a name, characteristics, goals, personal background
- Develop a small set of personas with one primary
Example Persona

Ginnie

**BACKGROUND**
- 15, Female
- Ongoing Private Education
- Ambitious
- Comfortable using technology to communicate

**MOTIVATIONS**
- Keeping in touch with her network
- Fashion/street cred
- Keeping up with peers.

**FRUSTRATIONS**
- Sad people trying to be 'friends' on Facebook
- Having to be in bed @ 11pm
- Being swamped in friends updates
- Missing important status updates

**Receives private tutoring in Maths and English as these are not her strong subjects. Enjoys playing for the school’s 2nd teams for netball and Lacrosse and is good at art.**

She loves recording her favourite shows: ER and Sun Valley High on Sky+ and spends some of her time on her Laptop that Daddy bought her watching videos on YouTube, downloading music, keeping up to date with her friends on Facebook and chatting via MS IM to her cousin who is at University in Leeds.

She loves Ugg boots and Abercrombie & Fitch and uses the Internet to shop and find the cheapest prices.

“**I want to easily hook up with my friends whilst watching TV**”
Example Persona

Jill Anderson

Bio
Jill is a Regional Director who travels 4-8 times each month for work. She has a specific region in which she travels, and she often visits the same cities and stays in the same hotel. She is frustrated by the fact that no matter how frequently she takes similar trips, she spends hours of her day booking travel. She expects her travel solutions to be as organized as she is.

Goals
- To spend less time booking travel
- To maximize her loyalty points and rewards
- To narrow her options when it comes to shop

Frustrations
- Too much time spent booking - she’s busy!
- Too many websites visited per trip
- Not terribly tech saavy - doesn’t like the process

Personality

<table>
<thead>
<tr>
<th>Extravert</th>
<th>Introvert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensing</td>
<td>Intuition</td>
</tr>
<tr>
<td>Thinking</td>
<td>Feeling</td>
</tr>
<tr>
<td>Judging</td>
<td>Perceiving</td>
</tr>
</tbody>
</table>

Motivations

- Price
- Comfort
- Convenience
- Speed
- Preferences
- Loyalties/Rewards

Brands

- Kayak
- Basecamp
- Outlook
- Enterprise
- IHG

Technology

- IT & Internet
- Software
- Mobile Apps
- Social Networks
Persona tools

- https://app.xtensio.com/
- https://uxpressia.com/
- http://personapp.io/
- https://www.justinmind.com
- ......
At last....the living persona! (Aer Lingus 🍀)
Data gathering for requirements!

- Interviews:
  - e.g. sample scenarios of use,
  - good for exploring issues
  - development team members can connect with stakeholders

- Focus groups:
  - group interviews, good at gaining a consensus view and/or highlighting areas of conflict
  - but can be dominated by individuals
Data gathering *for requirements!*

- **Questionnaires:**
  - often used in conjunction with other techniques, can give quantitative or qualitative data
  - good for answering specific questions from a large, dispersed group of people

- **Researching similar products:**
  - good for prompting requirements
Data gathering **for requirements!**

• Direct observation:
  — gain insights into stakeholders’ tasks
  — good for understanding the nature and context of the tasks
  — but it requires time and commitment from a member of the design team, and it can result in a huge amount of data

• Indirect observation:
  — not often used in requirements activity
  — good for logging current tasks
Data gathering for requirements!

Studying documentation:

— procedures and rules are often written down in manuals

— good source of data about the steps involved in an activity, and any regulations governing a task

— not to be used in isolation

— good for understanding legislation, and getting background information

— no stakeholder time, which is a limiting factor on the other techniques
Considerations for data gathering (1)

• Identifying and involving stakeholders: users, managers, developers, customer reps? union reps? shareholders?

• Involving stakeholders: workshops, interviews, workplace studies, co-opt stakeholders onto the development team

• ‘Real’ users, not managers

• Political problems within the organisation

• Dominance of certain stakeholders

• Economic and business environment changes

• Balancing functional and usability demands
Considerations for data gathering (2)

- Requirements management: version control, ownership

- Communication between parties:
  - within development team
  - with customer/user
  - between users... different parts of an organisation use different terminology

- Domain knowledge distributed and implicit:
  - difficult to dig up and understand
  - knowledge articulation: how do you walk?

- Availability of key people
Data gathering guidelines

- Focus on identifying the stakeholders’ needs
- Involve all the stakeholder groups
- Involve more than one representative from each stakeholder group
- Use a combination of data gathering techniques
- Support the process with props such as prototypes and task descriptions
Task descriptions

- **Stories**: a natural way for people to explain things
- **Scenarios**
  - an informal narrative story, simple, ‘natural’, personal, not generalisable
- **Use cases**
  - assume interaction with a system, step by step
  - assume detailed understanding of the interaction
  - show actions that are required to reach a goal
- **Essential use cases**
  - abstract away from the details
  - does not have the same assumptions as use cases
Task descriptions *(not in book)*

- User stories
from the perspective of the end user, what the user *wants to do* simple in structure:

*As a runner I *want* to track the miles I run each day so that I can have a good understanding of how much I have exercised.*
Scenarios and Personas

Figure 10.10 The relationship between a scenario and its associated persona

Source: http://www.smashingmagazine.com/2014/08/06/a-closer-look-at-personas-part-1/
“The Thomson family enjoy outdoor activities and want to try their hand at sailing this year. There are four family members: Sky (10 years old), Eamonn (15 years old), Claire (35), and Will (40). One evening after dinner they decide to start exploring the possibilities. They all gather around the travel organizer and enter their initial set of requirements – a sailing trip for four novices in the Mediterranean. The console is designed so that all members of the family can interact easily and comfortably with it. The system’s initial suggestion is a flotilla, where several crews (with various levels of experience) sail together on separate boats. Sky and Eamonn aren’t very happy at the idea of going on vacation with a group of other people, even though the Thomsons would have their own boat. The travel organizer shows them descriptions of flotillas from other children their ages and they are all very positive, so eventually, everyone agrees to explore flotilla opportunities. Will confirms this recommendation and asks for detailed options. As it’s getting late, he asks for the details to be saved so everyone can consider them tomorrow. The travel organizer emails them a summary of the different options available.”
Use case for travel organizer

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country’s name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.
8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.
11. The user chooses to print the requirements.
## Use case: written out

<table>
<thead>
<tr>
<th>Name: ATM withdrawal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: Actor wants to withdraw money from account</td>
</tr>
<tr>
<td>Actors:</td>
</tr>
<tr>
<td>• Customer</td>
</tr>
<tr>
<td>• Customer representative</td>
</tr>
</tbody>
</table>

**Use Case Relationships**

**Pre-Conditions**

**Basic Flow**

1. Use case begins when Actor presents ATM card
2. Validate customer:
   - **Business Rule: Customer must be Valid.**
   - **Business Rule: Customer pin must match.**
3. Determine if Customer has sufficient funds:
   - **Business Rule: Requested amount must be less than total amount in account**
4. Deduct amount from account
5. Distribute money
6. Use case ends with distribute receipt

**Post Conditions**
Use case diagram

Website Use Case Diagram

Site user:
- Search docs - full text
- Browse docs
- View events
- Log in
- Upload docs
- Post new event to homepage
- Add user

Webmaster:
- Add company

Download docs
- Preview doc
- Manage folders
Example essential use case for travel organizer

retrieve Visa

<table>
<thead>
<tr>
<th>USER INTENTION</th>
<th>SYSTEM RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>find visa requirements</td>
<td>request destination and nationality</td>
</tr>
<tr>
<td>supply required information</td>
<td>obtain appropriate visa info</td>
</tr>
<tr>
<td>obtain copy of visa info</td>
<td>offer info in different formats</td>
</tr>
<tr>
<td>choose suitable format</td>
<td>provide info in chosen format</td>
</tr>
</tbody>
</table>
Alternative courses (for travel organizer)

1 t/m 5…..

6. The user enters her nationality.
   
   If the country name is invalid:
   
   1. The system displays an error message.
   2. The system returns to step 3.

8. The system checks the visa requirements of the entered country for a passport holder of her nationality

   If the nationality is invalid:
   
   1. The system displays an error message.
   2. The system returns to step 6.

9. The system displays the visa requirements.

   If no information about visa requirements is found:
   
   1. The system displays a suitable message.
   2. The system returns to step 1.
Task analysis

- Task descriptions are often used to envision new systems or devices
- Task analysis is used mainly to investigate an existing situation
- It is important not to focus on superficial activities
  - What are people trying to achieve?
  - Why are they trying to achieve it?
  - How are they going about it?
- Many techniques, the most popular is Hierarchical Task Analysis (HTA)
Example Hierarchical Task Analysis

1. In order to buy a DVD
2. locate DVD
3. add DVD to shopping basket
4. enter payment details
5. complete address
6. confirm order

plan 0: If regular user do 1-2-5.
        If new user do 1-2-3-4-5.
Example Hierarchical Task Analysis (graphical)

Figure 10.15 A graphical representation of the task analysis for buying a DVD
DESIGN, PROTOTYPING AND CONSTRUCTION

Chapter 11
Overview

- Prototyping
- Conceptual design
- Concrete design
- Using scenarios
- Generating prototypes
- Construction
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
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 a 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
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’
Low-fidelity Prototyping

- Bv. papier of plastic of prototyping software
- Geen overdreven visueel detail
- Snel gemaakt, snel veranderd
Prototype: Lo-fi
# Prototyping

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

Table 11.3 Advantages and disadvantages of low- and high-fidelity prototypes
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
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
Wireframes
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
High-fidelity prototyping: 2 cases

1. Netsuky project
2. The EFB
Case: The Netsuky project:

*Software as an aid in conversational memory*

- Context: call centers
- In Europe 1% works in them (in USA 3%).
- Collaboration UU – Startup in Ireland

(deze case is geen tentamenstof)
High Fidelity Prototyping Case 1: Netsuky

Context: call centres & their software

- High turnover
- Only (phone) auditory contact
- Very anonymous
- Often “grumpy” conversations
- Routine tasks, many customers per day
- Monotonous
- Too few “sincere” commitments
- It must improve, change the tools!
High Fidelity Prototyping Case 1: Netsuky

Context: call centres & their software

- Diminish distance between human operator and the case he/she is working on by taking away some parts of task back to HUMANS, away from the system or machine. Make it such that an operator becomes more “owner” of a case.

- This part is generated by the operator (him/herself)! This part costs time, creativity and effort

- Rationale behind it: if something is self-generated it will leave stronger, more lasting and more meaningful memory traces. Exploit multimodality: text & images
High Fidelity Prototyping Case 1: Netsuky

Standard flow

1. Pickup phone, converse
2. Data entry, during and after (semi automated)
3. Send to backoffice
4. Next customer
High Fidelity Prototyping Case 1: Netsuky

Netsuky flow

1. Pickup phone, converse
2. Data entry, during and after
3. System demands 2 effortful creative episodes from user
4. Send to backoffice
5. Next customer
High Fidelity Prototyping Case 1: Netsuky

Netsuky flow
High Fidelity Prototyping Case 1: Netsuky

Netsuky vs. standard flow
High Fidelity Prototyping Case 1: Netsuky

Netsuky vs. standard flow: Experiment

Session 1
N=46
24 x netsuky version, 22 control version
27 x listening, do data entry task

Session 2 (3 days later)
N=46
Netsuky: see name and (self chosen) image
Control: see only the name
Reproduce: facts, details, moods and rate confidence
High Fidelity Prototyping Case 1: Netsuky

Standard version in experiment

You are logged in as: chris

Request: 

Name of caller: 

Save and go to next phone call

1. Mr. Scotland
2. Karen
3. Rotary furnishings
4. Miss Thompson
5. Forth Credit Union
6. Mr. George
7. Mr. Foyle
8. David
9. Mr. Morgan
10. Yvonne Stall
11. Cheryl A. Johnston
12. Miss Clark
13. Scott Snyder
14. Pete Scanlon
15. Our Lady of Victory Church
16. John Cruz
17. Grainne Doyle
18. Hazel
19. Margaret Gallery
20. Denis
21. Karen Morrison
22. Richard Walters
23. Robin Walker
24. Hazel Harper
25. Tracy Hartman
26. Karen Starfield
27. Sarah Maskill
Netsuky version in experiment
# Experiment session 2: recall

<table>
<thead>
<tr>
<th>Request</th>
<th>All facts</th>
<th>Mood etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mr. Scotland</strong></td>
<td>door handle be fixed in home.</td>
<td>Scotish man, in his 30s. Was anxious and a little worried as to when it would be fixed. Insisted it.</td>
</tr>
<tr>
<td><strong>Woman wanted to check balance.</strong></td>
<td>Needed to be resolved fast, for his elderly father. Customer tried to fix it himself, but</td>
<td></td>
</tr>
<tr>
<td><strong>Karen</strong></td>
<td>Was done easily.</td>
<td>American. Customer and Representive were both chaty, happy young women.</td>
</tr>
</tbody>
</table>

| **Mr. Scotland**         | Something to do with banks.                                               | Male, Scottish, 40ish years old, seemed generally fine. |
| **Karen**                | Transfer money                                                            | Female, 30ish, English, |

```
<table>
<thead>
<tr>
<th>Numbers</th>
<th>Numbers</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
How an interface behaves has a huge influence on our cognitive processes.

In certain situations, designers could take care that systems don’t do TOO much for the user (depending on task, context). Sometimes it is good to, counterintuitively, making a process a bit more effortful or demanding and less assisted.

Multimedia/ICT are everywhere. These findings could apply in a range of other applications, including educative software, or systems in which making mistakes has huge consequences.

This is a manifest for ourselves: there are things that we humans are best at 😊! And it makes it more fun too!
High Fidelity Prototyping Case 2: The EFB

The AVIOBOOK is a truly comprehensive and proven EFB software solution, designed for use across portable and installed hardware, for Windows and iOS platforms.

(deze case is geen tentamenstof)
High Fidelity Prototyping Case 2: The EFB
High Fidelity Prototyping Case 2: The EFB

(First: Contextual inquiry)
High Fidelity Prototyping Case 2: The EFB

(First: Contextual inquiry)
High Fidelity Prototyping Case 2: The EFB

- Look & feel al anwezig
- Functioneel (of simuleert dit geloofwaardig)
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
Prototyping: horizontal or vertical

Horizontal Prototyping in a Website

Vertical Prototyping in a Website
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
Also here: many tools!
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
Prototyping tools *(not in book)*

- Axure
- Invision

- **Prototypes**
- **Feedback**
- **Collaboration**
- **Workflow**
- **User Testing**
- **Boards**
INTERACTION DESIGN IN PRACTICE

Chapter 12
Overview

- Recap / the old stuff
- Agile UX
- Lean UX
- Design Patterns
- Customer journey mapping
Waterfall model
Waterfall model: downwards and not backwards
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Agile <-> UCD

- Design is too important to leave it to designers.

- Development is too important to leave it to developers.
Agile development

• Short (one to three week) time boxes 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
BDUF (Big Design UpFront)

Works well with

Mistakes …are expensive

In Agile: Avoid BDUF

Materials are cheap. Building is cheap. Changes are easy
Hoewel....?
“Enough storyboarding. Let’s shoot something.”
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 time boxing approach **does not support** long periods of user research
- User evaluations and some detailed work can be fitted within a time box
- Some user research can be performed in iteration 0 (zero), before implementation starts
- Ongoing program of user research
Old versus new style (1)

Waterfall UX (Bad!)

UX → Dev
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 (3)

Table 1: Comparison of heavy- and light-weight processes exemplified by a few chosen aspects of principles and methods.

<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>
</tbody>
</table>
Lean UX
Lean UX: 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
Agile vs. Lean

Sometimes used interchangeably……

Differences
• Roots of Agile are in production systems, Agile comes from creative and knowledge environments
• Lean has strong focus on an entire organisations whereas Agile is more famous for team-approach.

Commonalities
• Both are customer-focused approaches developed in response to the shortcomings of plan-driven methods
• Nothing is more important than the end result
• Continuous learning and adapting is central
• People are more important than tools
• Both are as set of principles
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?
Customer journey mapping
Customer journey mapping: many tools

uxpressia.com

smaply.com
Vragen?