OP482 - HTI Design Track A

Prototyping and evaluation

Wijnand IJsselsteijn

TM-MTI, IPO 1.37
tel: 4455
e-mail: w.a.ijsselsteijn@tue.nl
Aim of this class

- Evaluation & Prototyping in design cycle
- Overview of prototyping methods
- Overview of evaluation methods
  - Evaluation without users
  - Evaluation with users
Recall:

The user is just like me

Designer
User-Centred Design

- Design based upon a user’s
  - abilities and real needs
  - context
  - tasks/activities
- Prototyping as a method to involve the user in the design process
Product development phases

Phase 1: User Needs Analysis
Phase 2: Requirements Specification
Phase 3: Conceptual Design
Phase 4: Prototype, Development, and Test
Phase 5: Product Evaluation & Follow-Up
Prototype - *Webster* definition

- **Etymology:** from Greek *prototypon* archetype; *prototypos* original, primitive
  - an original model on which something is patterned: archetype, pattern
  - an individual that exhibits the essential features of a later individual or species: precursor
  - a standard or typical example of the essential features of a group or type: exemplar
  - a first full-scale and usually functional form of a new type or design of a construction
Architectural Prototypes

- Sketches, maps, artist’s impressions, mock-ups, ‘flythrough’ videos, computer-generated interactive models (VEs)
- Scaled-down model of the final building: mainly layout and aesthetics, minimal functionality
Prototyping in HCI

- Building a scaled-down version of an interactive system to collect information to guide its construction - a shared, concrete artefact

- Information gathering process
  - Acquire the information needed to successfully build a system

- Invaluable technique for iterative design
  - Expedite the iterative refinement cycle
Prototyping Techniques

- Stimulate creative thought & visualise ideas
- Consider design alternatives - explore design space
- Enable *early* user testing
- Anticipate usability problems
- Enable communication
- Get user and design team “buy-in”
Information from Prototyping

- Acquire the information needed to build a usable and useful system
  - Task specification
  - Interface functionality
  - Screen layouts and behavior
  - Design rationale
  - User feedback
Prototyping Tool Requirements

- Make information gathering process effective and cheap
  - Ease of use
  - Fast turn-around
  - Flexibility
  - Data collection capabilities
  - Executable prototypes
  - Team design
Prototyping for Whom?

- Primarily to identify user needs to HCI designer
- Software design in multi-competence teams
- Prototypes as artifacts for communication between HCI designers and
  - Users
  - Clients
  - Software design team
  - Graphic design team
  - Management
  - Other HCI designers
One Prototype for All?

- Central role of prototypes in communication and coordination
- Different team members → different roles → different activities → focus on different aspects of a prototype
- Appropriate prototype representation for
  - Audience (users, engineers, investors, …)
  - Design phase
When to Build a Prototype?

- Starting from task analysis or user feedback of initial design idea
- Goal of prototyping
  - Design validation, proof-of-concept
  - Basis for next evaluation
- Iterative refinement of prototypes during the development lifecycle
  - Starting with initial analysis
  - Until implementation and testing
Forms of Prototyping

**Early design**
- Brainstorm different representations
- Choose a representation
- Rough out interface style
- Task centered walkthrough and redesign
- Fine tune interface, screen design
- Heuristic evaluation and redesign
- Usability testing and redesign
- Limited field testing
- Alpha/Beta tests

**Low fidelity prototypes**

**Medium fidelity prototypes**

**High fidelity prototypes**

**Late design**
- Working systems
Lo-Fi Prototypes

- Low-fidelity ("off-line"): created quickly, usually in early stages of design and thrown away afterwards.
- Enable rapid iteration cycle, and more exhaustive exploration of design space.
- Less likely to constrain designers’ thinking or users’ evaluations
- Everyone can do them: participatory design
- Examples: paper prototyping, scenarios, Wizard of Oz, video prototyping
Hi-Fi Prototypes

- High fidelity ("on-line"): time-intensive, effective in later stages of design, partly reusable components
- Suitable for testing more complex, real-time functionality; dynamic visualisations
- Examples: computer animations, façade prototypes, interface builders, limited functionality product
Hi-Fi vs. Lo-Fi Prototypes

- Too long to build and change
  - weeks vs. hours
- You get feedback on ‘fit and finish’ issues but you need feedback on ‘big things’
  - Expressiveness and power of basic metaphor, flow of conversation, terminology
- Developers resist change - invested too much already in one specific design solution
- Hi-fi prototype may set expectations that are hard to meet
  - “… you are almost done …”
Overview: Limiting Prototypes

- **vertical prototypes**
  - includes in-depth functionality for only a few selected features
  - common design ideas can be tested in depth

- **horizontal prototypes**
  - the entire surface interface with no underlying functionality
  - a simulation; no real work can be performed

- **scenario**
  - scripts of particular fixed uses of the system; no deviation allowed

---

Paper Prototypes

- First prototype; quick and cheap
- Paper and pencil mockup of user interface
  - Rough sketches of the main interfaces and dialogs
  - Textual description of interface functions and relationships between screens (storyboards)
- Goal: brainstorming, first user feedback
Paper Prototypes

- Building prototypes on paper and testing them with real users
  - Construct paper prototype during first few days
  - Demonstrate behavior of an interface very early
  - Distill lessons from test observations

- Assumption: Quality is partially a function of the number of iterations and refinements

- Reasons for lo-fi prototyping
  - Still cheap to make changes
  - Maximizes number of design refinements before you must commit to code
  - Allows to try more ideas than with hi-fi prototypes
Components of a Paper Prototype

- Dialog boxes
- Main window
- Data on correction tape
- Menu bars

Marc Rettig: Prototyping for Tiny Fingers, Com. ACM 37, 1994
Pictive  Plastic Interface for Collaborative Technology  
Initiatives through Video Exploration

- Designing with office supplies
  - Multiple layers of sticky notes and plastic overlays
  - Different sized stickies represent icons, menus, windows etc.

- Interaction demonstrated by manipulating notes
  - New interfaces built on the fly

- Session videotaped for later analysis
  - Usually end up with mess of paper and plastic!
Pictive

- Can pre-make paper interface components

- buttons
- menu
- alert box
- combo box
- tabs
- list box
- entries
Paper Prototyping

Observer(s)
Observer taking notes on index cards

“Computer”
“Computer” with components laid out in order, for quick access

Facilitator
Facilitator, guiding user through tasks, prompting for user's thoughts

User
User, with lo-fi prototype in use

Facilitator pointed at interface
Scenarios

- Personalised, fictional - but possible - story with characters, events, products, and environments - e.g. ‘a day in the life’
- Help designer to explore ideas and the ramifications of design decisions in particular, concrete situations
- Multiple scenarios needed to reflect different situations
- Focus on social context of design
Scenarios

▪ An encapsulated description of:
  ▪ an individual *user*
  ▪ using a specific set of computer *facilities*
  ▪ to achieve a specific *outcome*
  ▪ under specified *circumstances*
  ▪ over a certain *time interval*

Snapshots & Storyboards

- **Snapshot**: single image (often cartoon) that captures a significant possible interaction
- **Storyboard**: sequences of snapshots which focus on the main actions in a possible situation.
  - Originally from film making - used to get the idea of a scene
Home

Level 1
Neuro Science

Level 2
Neuro Coloring Book
Name Brain Parts
Neuron
Brain Sock

Level 3
Sleeping & Dreams
Worksheet (Note-Taker)

Level 1
Level 2
Level 3

NeuroScience 4 Kids
BRAIN GAMES!
Wizard of Oz ...

The first “Chess Computer”

Built in 1770 by the Hungarian Baron Wolfgang von Kempelen
Wizard of Oz

- A method of testing a system that does not exist
- E.g. the voice editor, IBM 1984
Wizard of Oz

- Human ‘wizard’ simulates system response
  - interprets user input according to an algorithm
  - controls computer to simulate appropriate output
  - uses real or mock interface
  - wizard sometimes visible, sometimes hidden
    - “pay no attention to the man behind the curtain!”

- good for:
  - adding simulated and complex vertical functionality
  - testing futuristic ideas
Wizard of Oz Examples

- IBM: an imperfect listening typewriter using continuous speech recognition
  - secretary trained to:
    - understand key words as “commands”
    - to type responses on screen as the system would
    - manipulating graphic images through gesture and speech

- Intelligent Agents / Programming by demonstration
  - person trained to mimic “learning agent”
    - user provides examples of task they are trying to do
    - computer learns from them
    - shows how people specify their tasks

- In both cases, system very hard to implement, even harder to change!
Facade Prototypes

- Drawing editors with ability to specify input behavior
  - Looks like real application, but no application logic behind
  - Display canned data
  - Switch between screens

- Examples
  - Presentation programs like PowerPoint
    - Animation features, UI buttons for navigation to specific slides
  - Apple Hypercard
    - WIMP interfaces
    - Look and feel level
  - MacroMedia Authorware or Director
    - Interfaces with large graphical component
    - Interface functionality level
Interface Builders

- Interface construction tools
  - Linked into application to produce industrial strength implementation
  - E.g. Visual Studio (C++, Basic), Delphi, ...

- Excellent for showing look and feel
  - Extensive control over interface appearance – horizontal prototype
  - Partly re-usable in final product

- Limited functionality
  - Vertical functionality can be added selectively through additional programming

- Constrained to widget library
## Overview Tools: Pros/Cons

<table>
<thead>
<tr>
<th>Tool Category</th>
<th>Paper</th>
<th>Facade Tools</th>
<th>Interface Builders</th>
<th>Actual Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Use</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>--</td>
</tr>
<tr>
<td>Fast Turn-around</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>--</td>
</tr>
<tr>
<td>Flexibility and Control</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Executable Prototypes</td>
<td>--</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Team Design</td>
<td>+</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Summary: Prototyping

- Context: usability engineering lifecycle (multiple iterations), user-centred design
- Prototyping
  - allows users/development team to react to the design and suggest changes
  - integral part of *evaluation* – early user involvement
  - low-fidelity *vs.* high-fidelity
- Many approaches, methods and tools
  - Vertical, horizontal and scenario prototyping
  - Paper prototyping, storyboards, Wizard of Oz
  - Software prototype tools, user interface builders
Evaluation
Evaluation methods

- Evaluation without users:
  - expert reviews, heuristic evaluations
  - psychological modelling

- Evaluation with users:
  - observation and monitoring
    - field studies, ethnography
  - users opinions (more or less participative)
    - surveys, focus groups, probes, participatory design
  - experiments and benchmarks
Evaluation: without users

- **Aim:**
  - generate list of usability problems
  - predict usability problems

- **Reasons:**
  - limited / no access to users
  - too expensive and/or time-consuming
  - seriousness of consequences of errors
  - standards conformance
Heuristic evaluation

- Compare design against principles
- For example:
  - Error prevention
  - Flexibility and efficiency of use
  - Help users recognize, diagnose, and recover from errors
Prevent errors

- Disabled functions should appear as such.
Flexibility and frequency of use

- Accelerators speed up the interaction for the expert user
- Allow users to tailor frequent actions.
- Example: incorrect assumptions
Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language
More Error Messages

- An internal error occurred. The BeyondMail internal error code is displayed above.

- Unexpected Error. Please investigate.

- Are you sure you want to delete 'Ridges'?

- System Error &H80004005 (2147467259). Unspecified error

- This item doesn't belong here

- The operation completed successfully.
Recognition rather than recall

- Provide only limited number of options
Heuristic evaluation: pros and cons

- Good points:
  - useful as a checklist when developing UI
  - useful for ‘quick and dirty’ evaluation

- Not so strong points
  - the rules are a mixed bag (no underlying theory)
  - sometimes they are mutually contradictory
  - ambiguous, hard to interpret
"Discount" usability evaluation

- Combining empirical usability testing and heuristic evaluation:
  - write scenarios,
  - users ‘think aloud’
  - change scenarios
  - apply heuristics to scenarios
Walkthroughs

- Walk-through of cognitive and operational actions
- Track goals of users
  - compare user goals, with goals necessary for interaction with product
- Track actions of users
  - compare users expectations with actual product behaviour
A cognitive walkthrough requires that:

- The interface has been documented thoroughly (this does not mean that there has to exist an implemented interface)
- Extensive tasks have been identified and described in detail
- The knowledge and experiences of the end users have been identified and described
Walkthrough: pros and cons

- **Pros:**
  - Systematic comparison of system match user goals and actions
  - Based on psychological theory

- **Cons:**
  - only predicts effectiveness and efficiency (not about acceptability, pleasure, etc.)
  - fairly labour intensive
Modelling techniques

- Modelling (typical) user behaviour: e.g., GOMS, ACT-R
- Using task structure to analyse execution of task
- E.g. to determine time to fulfil task or to determine number of keystrokes needed
Modelling: pros and cons

- **Strong points:**
  - can make relatively good predictions about task performance

- **Not so strong points**
  - level of analysis too high: based on ideal user
  - only predicts effectiveness and efficiency (not about acceptability, pleasure, etc.)
  - very labour intensive
Evaluation: with users

- **Aims:**
  - to explore design space and obtain design inspiration (participatory design)
  - to understand users’ needs and context -> formulate requirements
  - to obtain user feedback on early and/or later prototypes to ensure optimal user experience (formative evaluation)
  - to compare designs against target requirements or other designs (summative evaluation/ benchmarking)

- **Some underlying reasons:**
  - to obtain wide-spread user acceptance and create a commercially successful product
  - to enable cost-savings in the engineering phase
  - to limit after-sales headaches
  - a shinier, happier world 😊
Interviews

- Users needed: ~5
- Lifecycle stage: Task & environment analysis, early design stages
- Main advantage: Flexible, in-depth attitude and experience probing
- Main disadvantage: Time-consuming. Hard to analyse and compare.
- Variations: contextual inquiry (e.g. tech-tours), scenario’s, guided speculation
Interviews

- Structured versus unstructured
- Open-ended versus closed questions
- Designers’ versus users’ world
  - contextual inquiry
  - ethnography
Contextual enquiry

- Interviews in work context
- Work objects available
- Easier to address realistic issues

- but: Interruptions
Examples contextual enquiry

- Early in process: determine requirements for information sharing space
- Late: Evaluate how product ends up being used in context of use
Structure of interview

- **Unstructured or in-depth interview:**
  - interviewer develops an interview guide
  - questions are formulated within scope of guide
  - possibility to pursue interesting facts

- **Structured interview:**
  - Pre-determined set of questions
  - Uniform information, comparability of data
Interview advantages

- More appropriate in complex situations
- Collects in-depth information
- Information can be supplemented with impression of interviewees (non-verbal info)
- Questions can be explained
Interview disadvantages

- Time-consuming and expensive
- Quality of data depends on
  - quality of interaction
  - quality of interviewer
- Bigger chance of bias by interviewer
Focus Groups

- Users needed: 6-9 per group
- Lifecycle stage: Task analysis, user involvement
- Main advantage: Spontaneous reactions and group dynamics
- Main disadvantage: Hard to analyse. Low validity. Social demand characteristics.
Examples focus groups

- Early: discuss context of use of speech control for consumer products, to understand requirements for new product

- Late: show concrete product, and discuss possible uses, advantages and disadvantages
Questionnaires

- Users needed: At least 30
- Lifecycle stage: Task analysis, follow-up studies
- Main advantage: Finds subjective user preferences. Easy to repeat (large groups).
- Main disadvantage: Design of sensitive, reliable, and valid questionnaire is challenging. Pilot work needed (to prevent misunderstandings). Response bias.
Examples questionnaire

- Early: information about user groups, activities, frequency of use, etc.

- Late: information about questions specific to new product, having been used, e.g. shavers: easy of use, smoothness after shaving, context of use
Questionnaire advantages

- It is less expensive than interviews
- It offers greater anonymity of subjects
Questionnaire disadvantages

- Sample bias – who will send it back?
- Restricted scope and depth – no further probing possible
- Questionnaire development is time-intensive: item construction, scale construction, establishing reliability and validity - pilot tests required
Probes

- Users needed: ~5 clusters (e.g. families)
- Lifecycle stage: environment analysis, early design stages.
- Variations: cultural probes (Gaver), technology probes (InterLiving)
Probes

- Probes handed out in user session
- Information returned over time
Probes

- Postcards with questions

- Maps with related questions
Probes

- Diary (with camera)
Observations

- Users needed: 3 or more
- Lifecycle stage: Task and environment analysis, follow-up studies
- Main advantage: Ecological validity; reveals users’ real tasks. Suggests functions and features.
- Main disadvantage: No experimenter control. Intrusive. Experimenter can affect user behavior. Time-consuming data analysis.
- Variations: participatory observation
Observations

- What to observe?
  - Which activities? E.g.
    - Representative sample
    - Only new functionality/tasks
    - Global use

- Where?
  - Lab, field study

- When?
  - Sampling..
    - Learning phase, and/or expert use
    - Once, or more observations over time
Observations

- How? Non-obtrusive..
- What? Times, errors, …
- Interacting with what? Final product, prototype
Observation considerations

- Observing users in lab
  - Giving them 10 tasks
  - Measure: errors, time, ‘confusion’
  - Controlled set-up
- Observe users at home
  - Study natural behaviour
  - Log their actions, using software
  - Better understanding of real issues
Types of observations

- Participatory observation:
  - researcher participates in activities, e.g. with or without others knowing

- Non-participatory observation:
  - non-intrusive
  - less chance of bias
Thinking aloud

- Users needed: 3-5
- Lifecycle stage: Formative evaluation, iterative design
- Main advantage: Pinpoints user misconceptions. Cheap test.
- Main disadvantage: Unnatural for users. Hard for users to verbalise.
- Variations: peer tutoring, co-discovery
Experiments and user tests

- Users needed: At least 10+
- Lifecycle stage: Competitive analysis, benchmarking, final testing
- Main advantage: Controlled study. Quantitative data. Results easy to interpret and compare. Replicable.
- Main disadvantage: Low generalisability. Tasks often artificial and restricted. Time-intensive.
User population

- ‘All’?
- or a few?
User representation

- Expert?
- Or real user?
Timing of involvement

- Early?
- And/or late?
Location of analysis

- Laboratory?
- In context?
Examples usability labs

- IPO UseLab
Examples usability labs

- PHILIPS HomeLab