Handheld Computer of the Future
- KidsTeam

Honors 208W
Fall 2011

Determining What the User Wants and Constructing Low Fidelity Prototypes

Task and User Centered Design

When designing new hardware and/or software, it is important to keep two types of things in mind:

- What are the tasks that people need to be able to accomplish?
- Who are the users, what are their abilities and experiences, how will they interact with your new tool to accomplish their desired tasks?

Task Centered Design (I)

Requirements analysis.

- Who would use your system?
- To do what?

Identify potential users and their tasks.

- Users: Get in touch with real people who are potential users of the system you are designing.
  - Possibly in general categories.
  - Possibly in very specific niche markets.
- Realistic example tasks: Concrete, detailed examples of task scenarios.
  - Everyday tasks.
  - Occasional yet important tasks.
  - Occasional and relatively unimportant.

Task Centered Design (II)

Design your system.

- Decide which users and tasks you will support. It might not be practical to design a system to support each and every task and/or user that you discovered in the previous stage.
- How will the hardware/software appear to the users? This is what the user first sees – it needs to invite use.
- What will each step through a given task look like? There should be a natural work flow as the user accomplishes their task.

Evaluate your design.

- Use the tasks examples to walk through your design to evaluate whether it will be usable.

Developing Task Examples (I)

Imagine you have a store that plans to have a catalog station where customers browse for products, then order items from a clerk.

- Scenario: A man carrying a demanding toddler buys an umbrella stroller (red is preferred but blue is acceptable), pays for it in cash, and uses it immediately.

What interface implication can we extract from this?

Developing Task Examples (II)

A good example scenario will:

- …say what the user wants to do, but not how they would do it. You do not want to tie any interface assumptions into the scenario description. This will allow you to compare different approaches without preconceptions.
- …be very specific. It needs to be clear exactly what the user wants to be able to do. It needs to present what "input" the user will need to be able to make (though not necessarily how they will provide it).
- …describe the complete task. These aren't meant to be lists of the individual things that are done in general. We want to be able to see the big picture, not just unlinked sub-goals.
- …identify the user. By knowing more specifics about the user, we can do things such as design based on what the user will already know how to do.

Performing a Walk-through

Before we build anything, we should evaluate whether our design plan makes sense on paper.

Select one of the (many) scenarios you have assembled and go through accomplishing the described task step by step.

- Is the motivation clear at each step?
- Can you expect the user to know what to do at each step with the anticipated level of training?

User Centered Design (I)

Notice that the user is always in our thoughts. Good design will consider the user's:

- Abilities
- Needs
- Context
- Tasks

We assume that:

- …the result of good design is a satisfied customer.
- …the design process is collaborative and incremental.
- …we are in touch with example users throughout our design, evaluation, and implementation process.

User Centered Design (II)

There are many ways in which users can be involved.

- Talk to potential users (seems obvious but isn't).
- Interview users more formally (find out more about their culture, expectations, abilities, surroundings).
- Explain what you are planning and welcome comments, criticisms, suggestions for change.
- Have them try "beta" versions early enough that changes can still be made (not simply shaking out the coding bugs).
- Have them be part of the design team!

Low Fidelity Prototyping

When building hardware and software, much of the design can happen without touching anything more high-tech than a pencil and pad.

A paper mock-up of an interface's look and feel can be shown to users before any investment in the actual development of the software/hardware.

- This is quick and cheap to create and easy to modify with the user. You can even design a few and compare/contrast them and merge ideas.
- It doesn't look like a product yet, so users can be more comfortable suggesting radical changes in the design.

Lo-Fi Tools

Paper/Pencil sketches of the general outward appearance convey the general ideas. This can allow us to focus on the high-level concepts.

Storyboards can be created to present snapshots of the interface and specific stages in the user interaction towards an actual goal. This can give a better feel of the progression of the workflow.

Can use techniques which incorporate sketches, sticky notes, transparencies, etc. with which we can simulate how the interface would actually behave without having to build anything. This can be more flexible in a walkthrough – consider a user that says "I'd click on that" to accomplish something and points to something that we thought was just decoration – you could move to the next screen or overlay anyway (something you couldn't do with a "real" implementation).

Some resources:

http://www.hcibib.org/tcuid/
http://en.wikipedia.org/wiki/User-centered_design
http://www.scis.nova.edu/~councilc/mmis660/pictive.htm