Image credit: http://worrydream.com/ABriefRantOnTheFutureOfInteractionDesign/
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 creation to accomplish their desired tasks?
Tasks and Users

Requirements analysis.

– Who would use your system?
– To accomplish what?
Real Users and Tasks

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 scenario examples.
  - Everyday tasks.
  - Occasional yet important tasks.
  - Occasional and relatively unimportant.
What everyday object did Norman say was often hard to use?

- 1. Key
- 2. Door
- 3. Flashlight
- 4. Paperclip

0% 0% 0% 0%
A version of which of the following was presented as unusable by Norman?

1. Bike
2. Chair
3. Candle
4. Wrench

0% 1. Bike
0% 2. Chair
0% 3. Candle
0% 4. Wrench

0 of 5
Think about the user...

The user should always be in your thoughts.

Good design will consider the user’s:

– Abilities
– Needs
– Context
– Tasks
Involve the Users

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).

Can even have them be part of the design team!
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.
Rettig advocates which of the following?

0% 1. Using your own product
0% 2. Building low fidelity prototypes
0% 3. Beta testing with early versions
0% 4. Having efficient algorithms

0 of 5
**Walk through your design**

Before you build anything, evaluate on paper.

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

For each scenario you have, 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?
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 design a few, 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.