Task-Centered Design
User-Centered Design

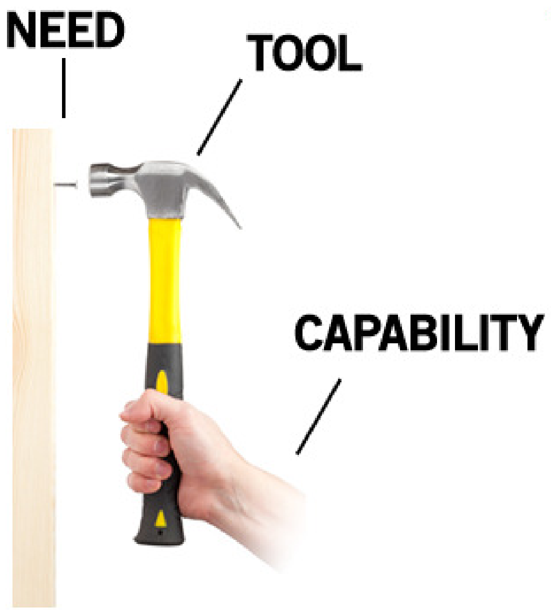


Image credit: http://worrydream.com/ABriefRantOnTheFutureOfInteractionDesign/

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.

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 things 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.

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.

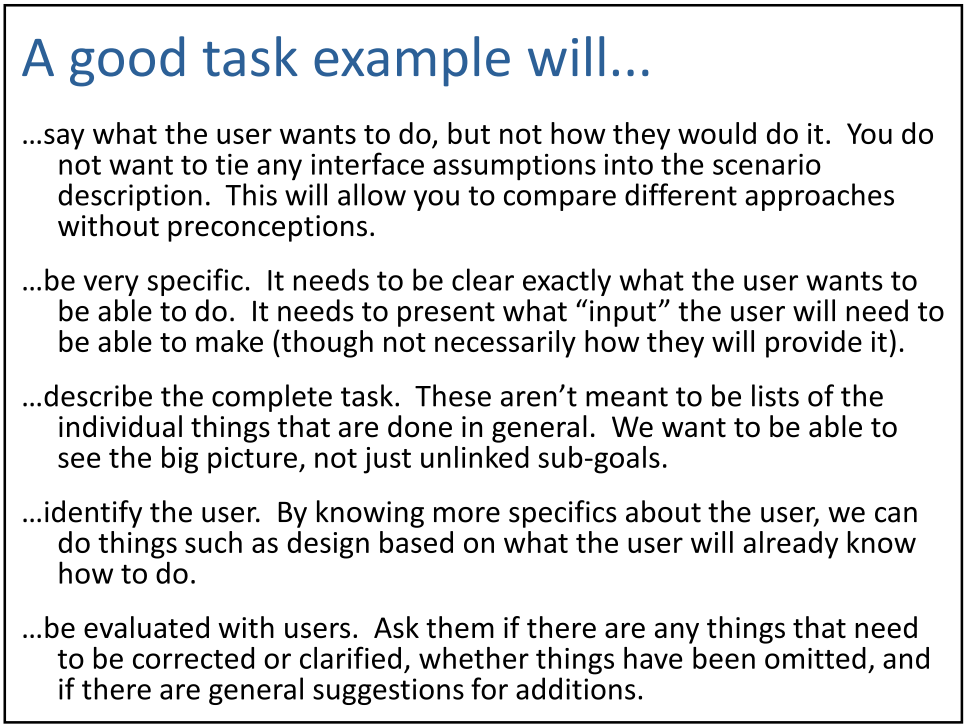
- Is the motivation clear at each step?
- Can you expect the user to know what to do at each step with the anticipated level of training?

An Example Task Scenario

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?



A good task example 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.

…be evaluated with users. Ask them if there are any things that need to be corrected or clarified, whether things have been omitted, and if there are general suggestions for additions.