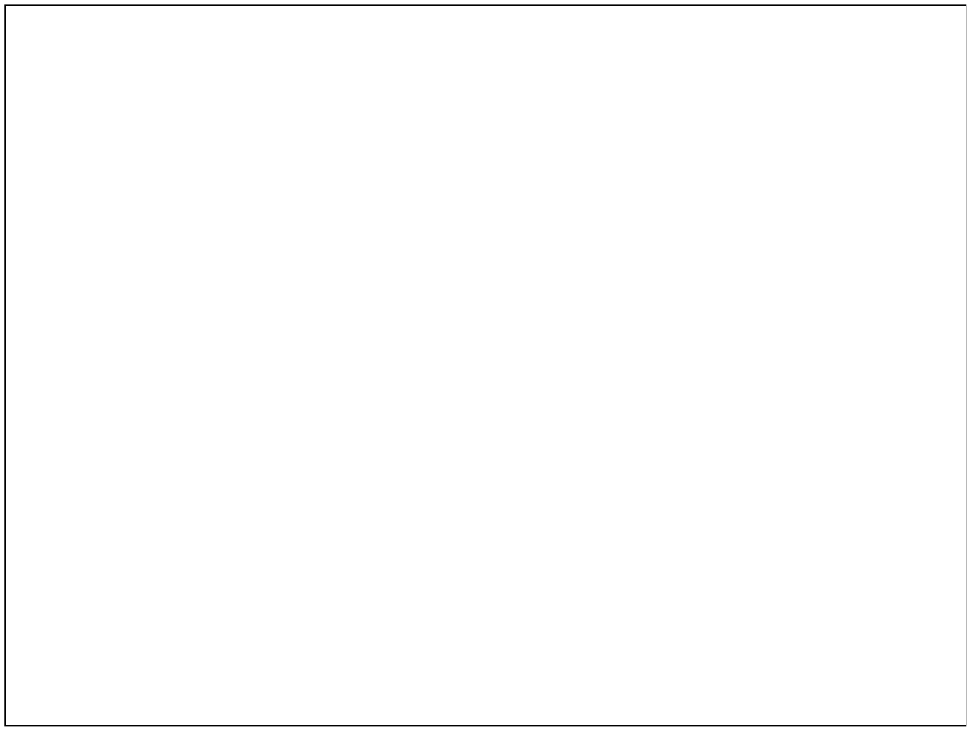
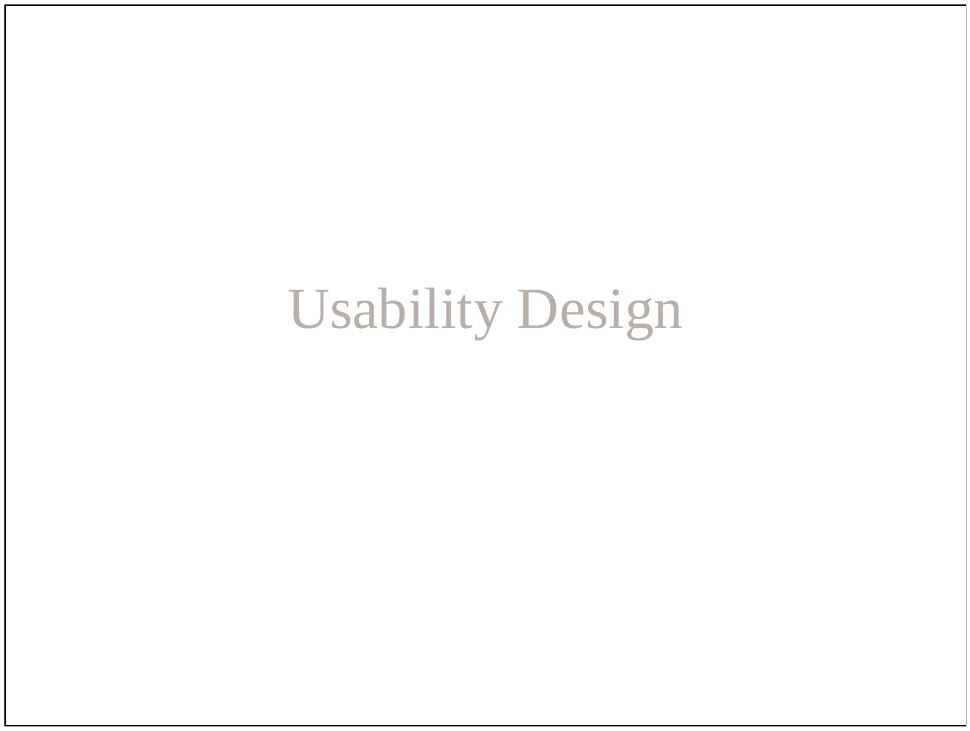


Use the cards to control the lecture…



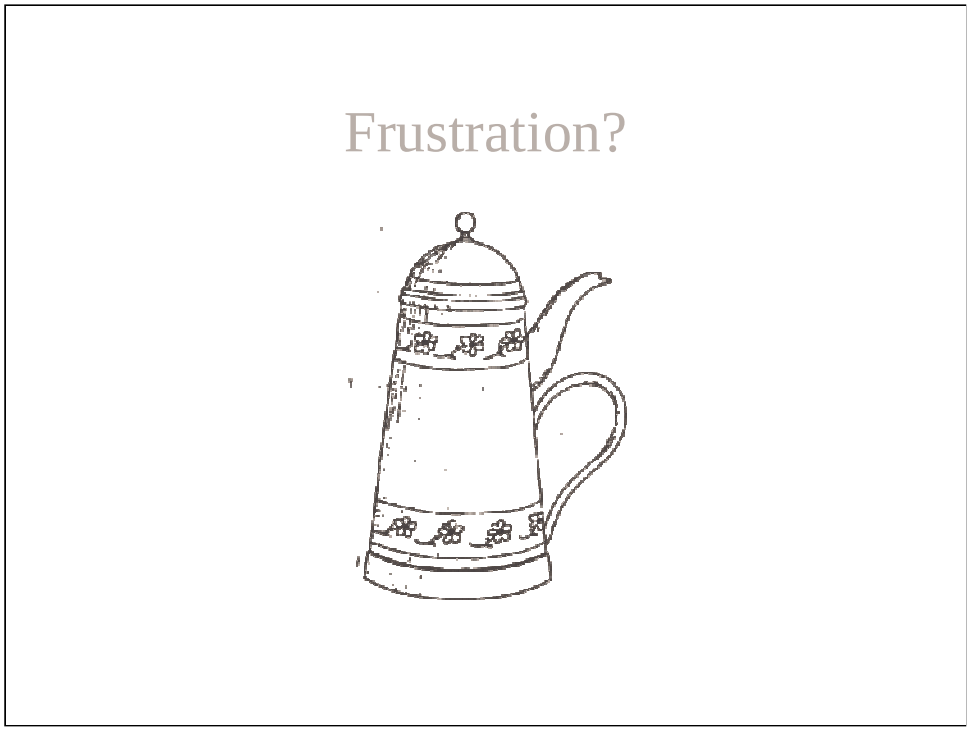
We’re going to talk about usability design.

Now, to be honest, right up front, I really don’t know much about usability design. But from what I can tell, nobody really does. Well, actually, a very small group of people know a great deal about this, but somehow there are still a lot of frustrating things out there.

I’ve tried to glean a few of the principles of usability design so that we here in this room might try and take a small chunk out of this huge problem.

Everything has an Interface

To start off with, it came as a small revelation to me that *everything* has an interface – doors, pens, telephones, maps, clocks, software, software *libraries*, even chairs or lectures. These are all things that we interact with all the time.



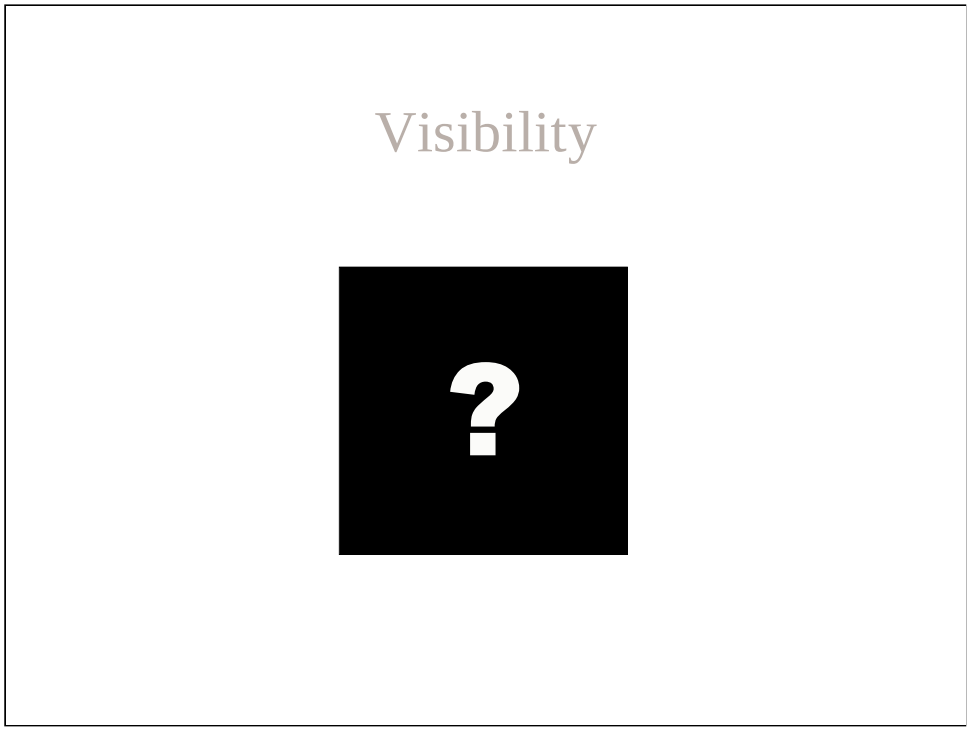
Unfortunately, many things that we interact with on a daily basis are poorly designed – some things are downright pathological. Let's think of some things that we've used recently that has frustrated us and let's steer clear of computers for now – that's just too easy!

…

What about items that have been easy, intuitive and a joy to use?

…

OK. So what separates the items that are good from those that are bad? Well, people who think about this a lot have come up with some broad overarching principles about well designed objects.



The first principle we're going to talk about is visibility. This might seem obvious, but to have to make visible, or some how expose the controls for a thing that is meant to be interacted with. A user should have to figure out what the controls are in order to use something. Moreover, if a control is used there needs to be some sort of visible indication that something has been done. I should never have to say " is this on?" This second aspect of the visibility principle is known as feedback, and could constitute a principle all its own.

Example

So here's an example. This pair of double doors (from the Saigon Deli in the ID) makes the controls really visible. The metal plate on the right door and the handle on the left door indicate that there is something to be controlled here. But what do these controls do?

Affordance

This brings us to the principle of Affordance. Pretend that you can't see the crack between the two doors. Which door would you think swings outward and which door do you think swings inward?

Affordance

How about now?

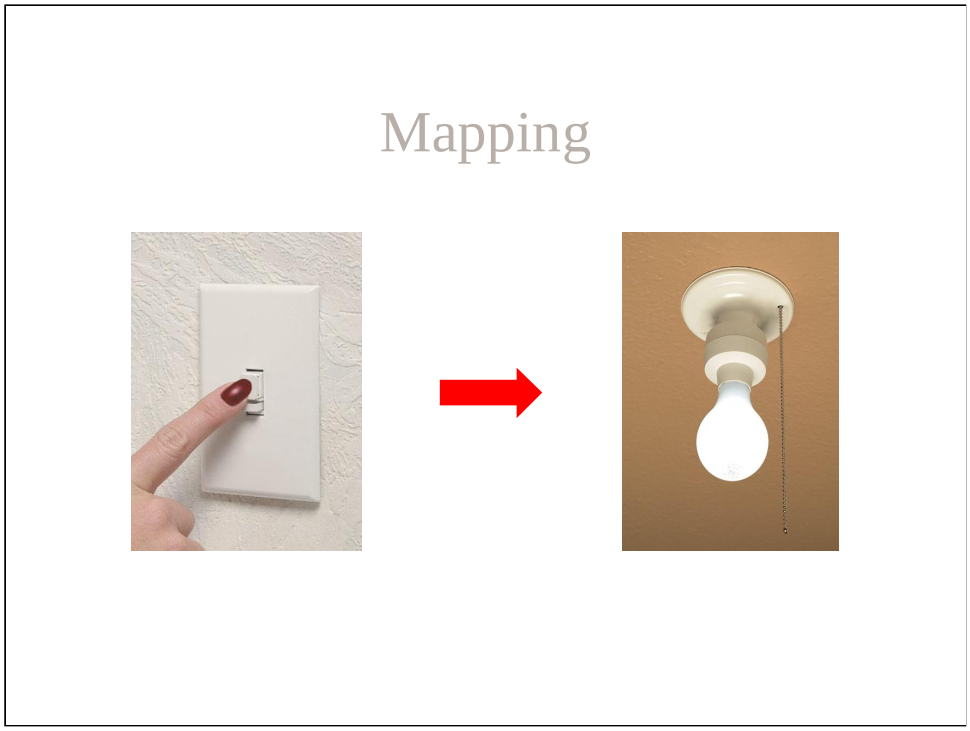
Affordance

And now?

It's probably obvious from my shoddy photoshopping that this is actually the real photo. What's wrong with this picture!?

Certain objects just lend themselves to certain actions. Knobs are for twisting, buttons are for pushing, handles for pulling, etc. Why certain objects imply certain actions is a complex topic, but what is important for our purposes is that controls should imply what should be done with them. By their very nature controls should tell the user how they can be manipulated. This is the essence of affordance.

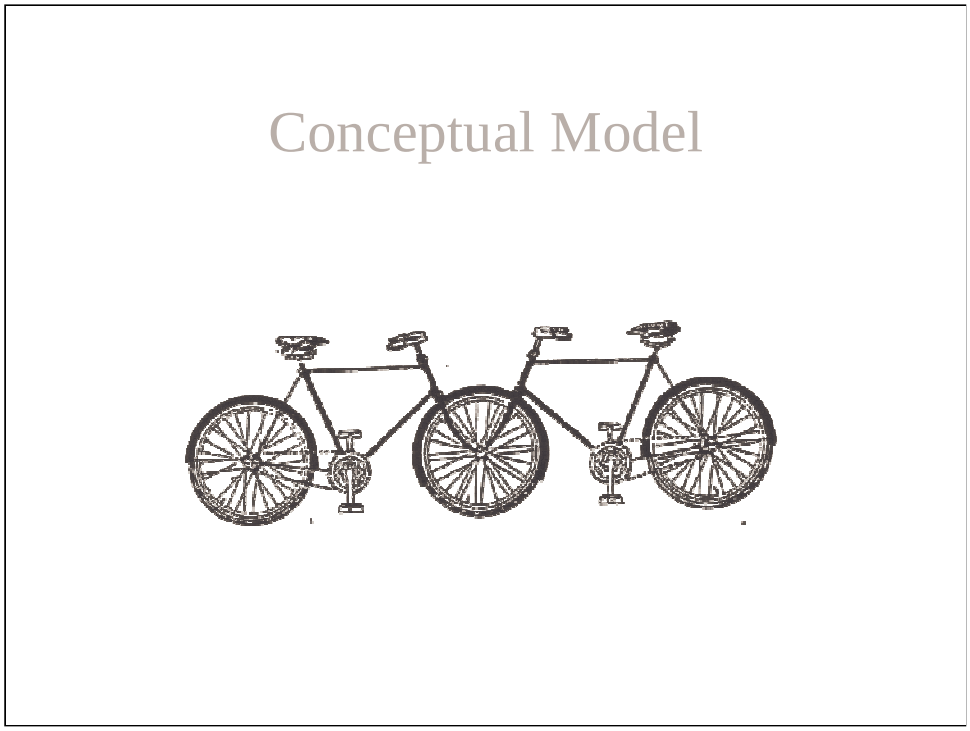
So fine, usable objects have visible controls that naturally imply how those controls can be twiddled. But what will happen when I twiddle?



This brings us to the concept of mapping. What does this term mean to you?

“Mapping” is technical term for the relationship between objects. In the context of usability, mapping usually refers to the relationship of some control to some behavior. Usable objects make this mapping clear or at least easy to learn. Frustrating objects obscure the mapping. What do youn think of the mapping pictured here? Is it easy? Is it natural, obvious, etc?

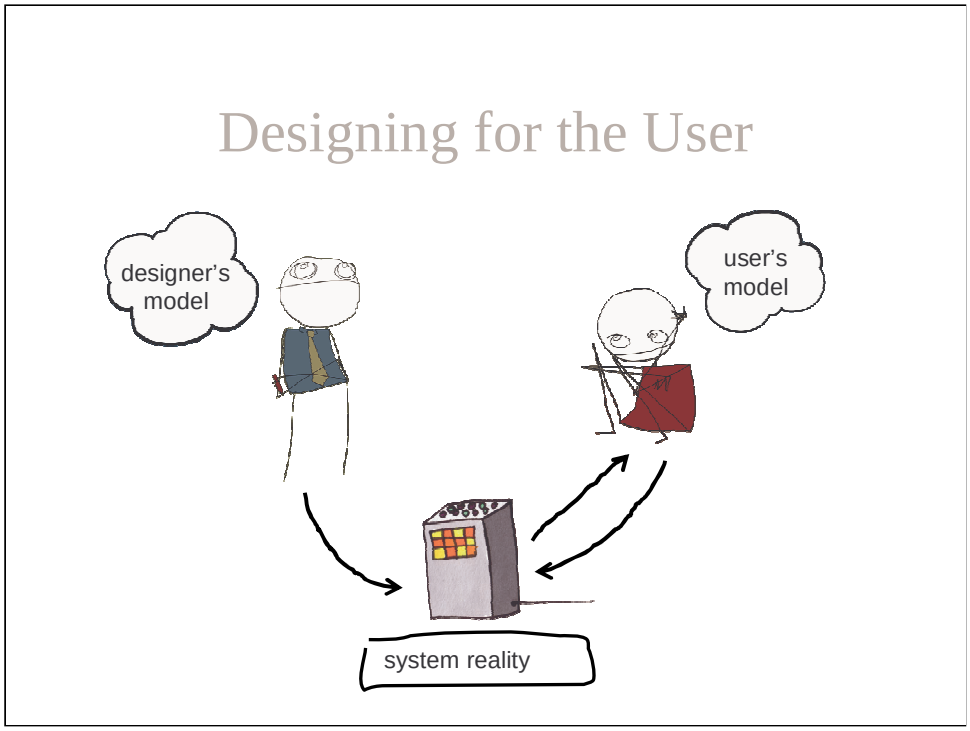
What happens when there are more functions than controls?



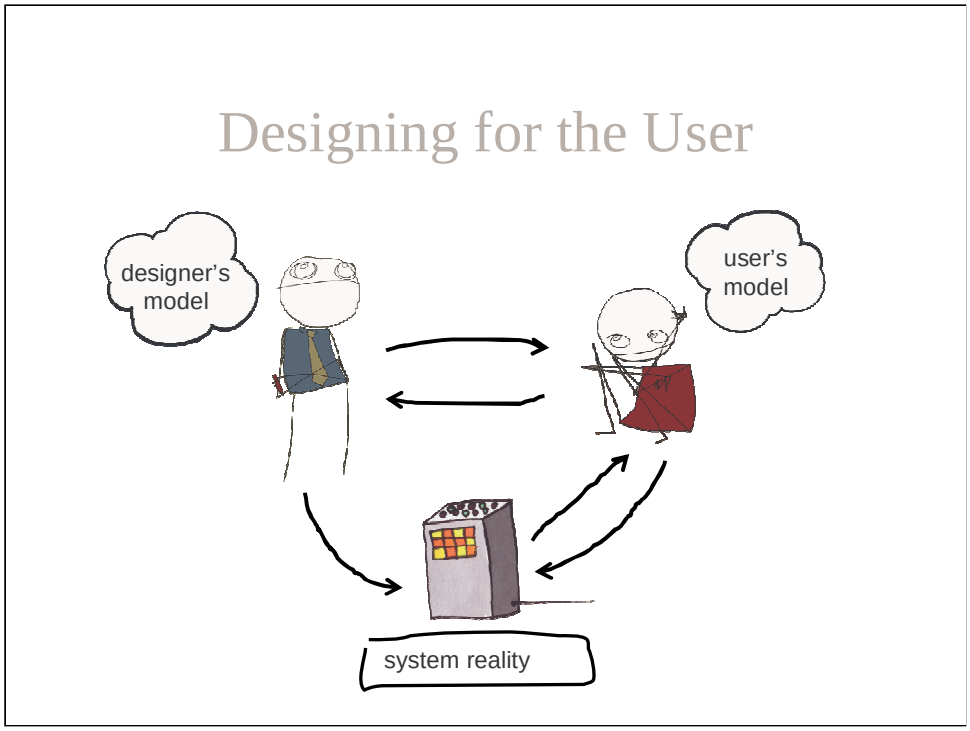
OK, so consider this machine. From what we've talked about so far, I would say this is fairly usable. Its controls are visible (handlebars, pedals, seats), they afford their function fairly well, and it's pretty easy to discern the mapping between each control and what it is meant to do. But is it usable? Probably not. Why? Well, between our experience and basic mechanical understanding, it seems clear that this isn't going to work. The problem is that we know this device won't work according to the conceptual model we've formed about it in our head given the visibility, affordance, and apparent mappings of the controls.

Yet it was presumably designed to be used? How are we do deal with this disconnect? The problem is that there appears to be a difference between the designer's model and the user's model. Good design brings these into confluence.

So how do we do this?



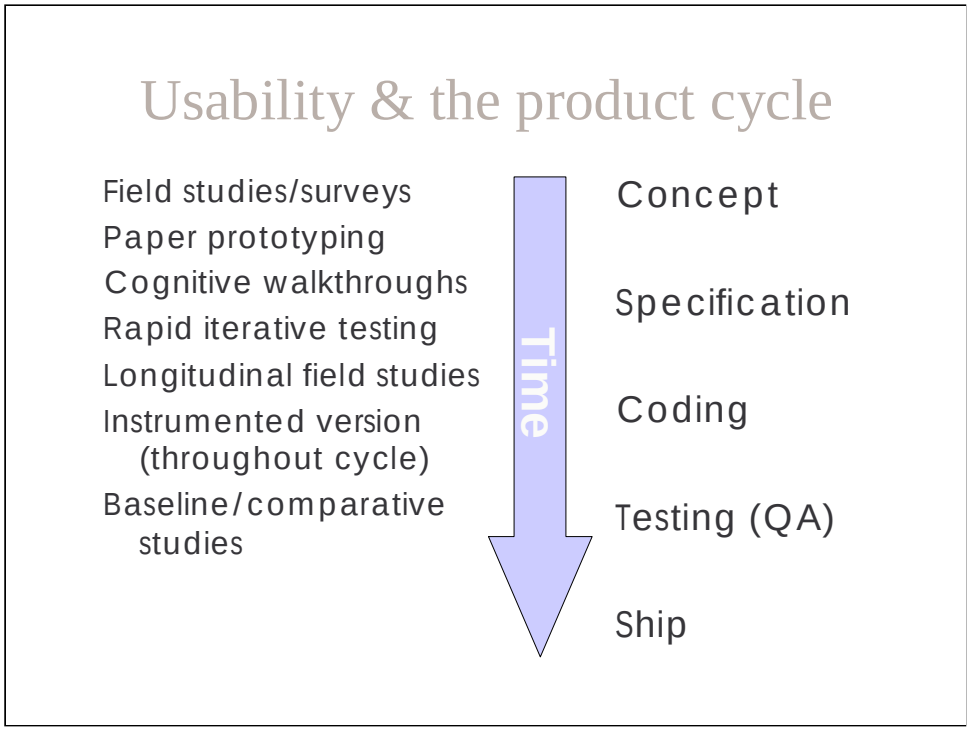
The traditional way that interfaces have been designed is to have the designer design it, and the user use it. End of story. But this causes problems since the only way for the user and the designer to communicate is through the system. If the system does not make the designer's conceptual model clear and consistent, then the user will have the wrong mental model and problems will ensue.



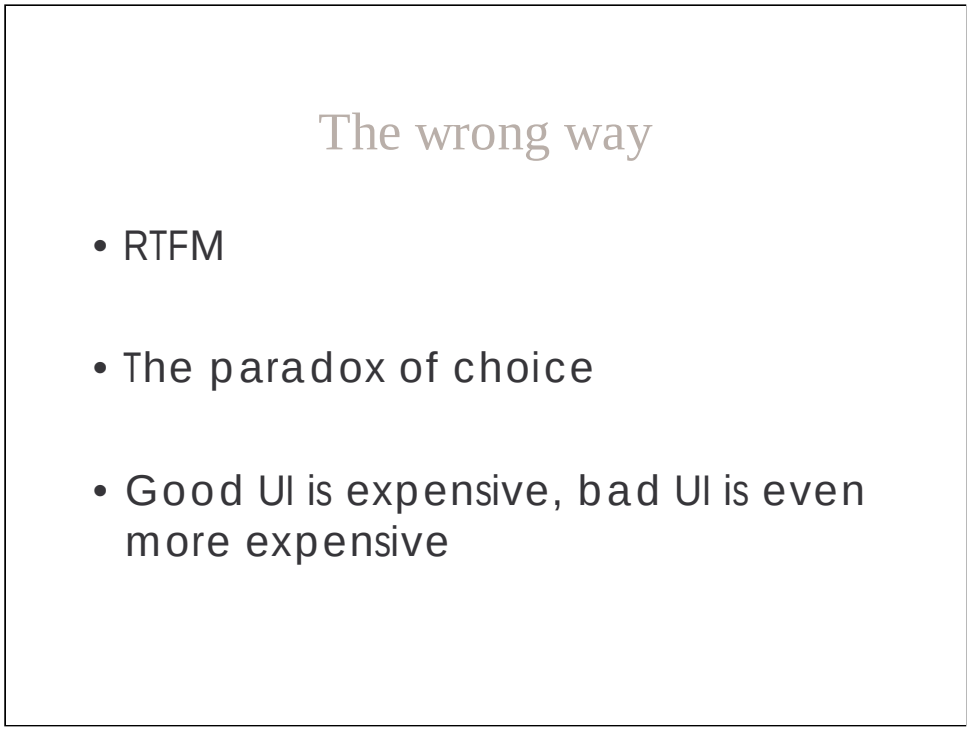
What we need is to bring the designer and user into communication before (or as) the system is being built.

Participate in product planning & design

- Review current user problems & needs
- Plan direction for User Interface (UI) in next version
- Review & coordinate on research efforts
- Review (and sometimes own) product specs
- Spec for user workflow.



We don’t use the traditional waterfall development process at Microsoft: iteration is key



If a user can't figure out how to use my tools then they should read the supporting documentation.

Users like choices.

I can not make it simpler because users need all the choices.

I'll impress the world with my all-in-one UI.

Different UI for different users: GUI, CLI, WMI, APIs

We can fix it if people complain.

Very expensive to fix later.

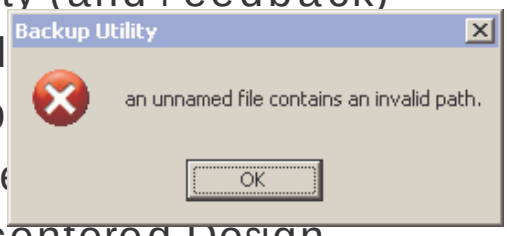
Beware of the *legacy UI phenomenon.* Legacy UI is hard to shed (QWERTY)

Recap

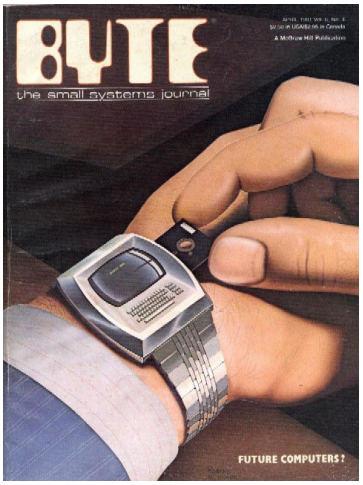
- Visibility (and Feedback)
- Affordance
- Mapping
- Conceptual Models
- User-centered Design

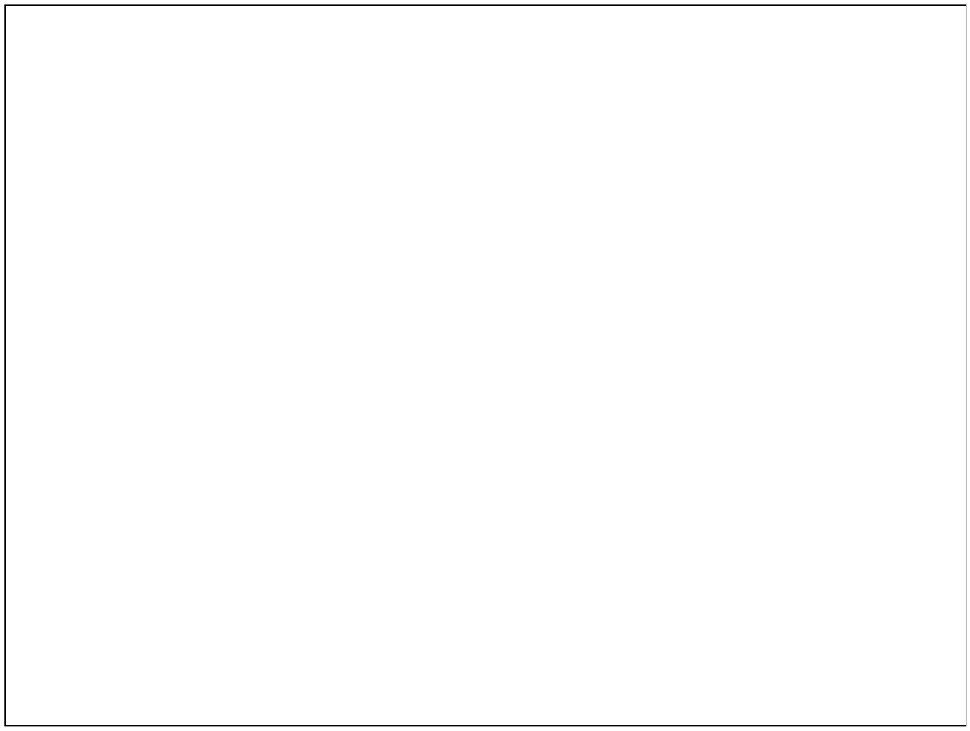
Recap

- Visibility (and Feedback)
- Afford
- Mapp
- Conce
- User-centered Design



Heaven Help Us…





Usability Questions

- What user needs can the product satisfy?
- What user needs does the current product miss?
- What design will solve those problems?
- How can the user succeed at task *X*?
- What can users do with the product?
- Is this product enjoyable to use?
- What can our competitor's product do?

Usability Activities

- Conduct iterative lab studies of products in development.
- Cognitive walkthroughs
- Customer visits--study users in real-world locations.
- Write/review/contribute to UI specs
- Engage with industry groups on common ease of use issues.
- Competitive evaluation.
- Surveys
- Statistical consultation and analysis.