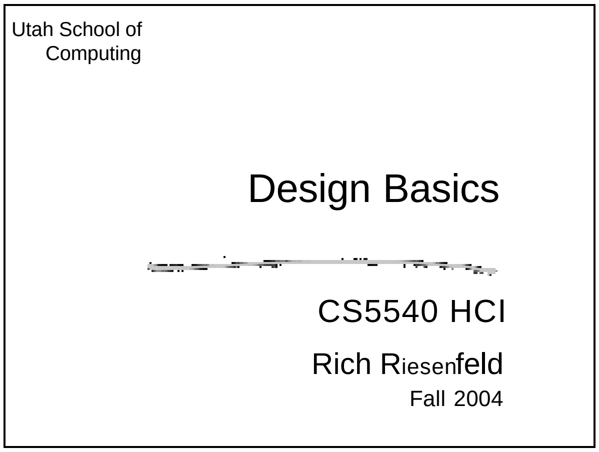
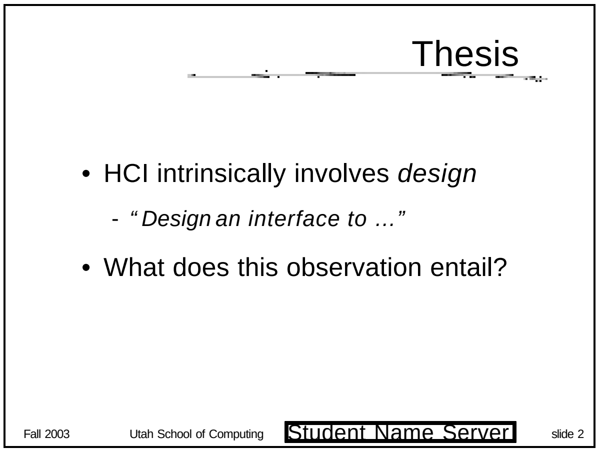


Utah School of
Computing

Design Basics

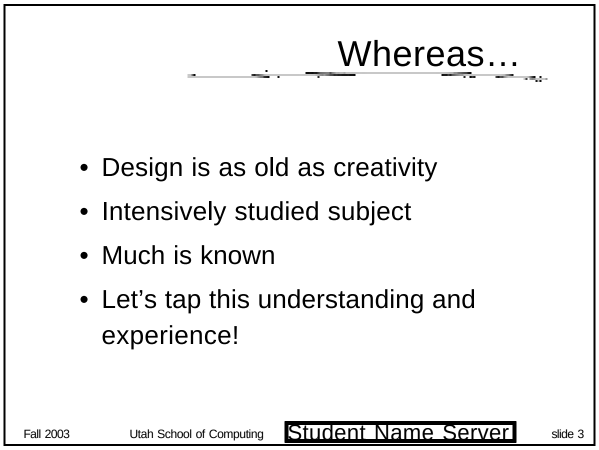
CS5540 HCI

Rich Riesenfeld
Fall 2004



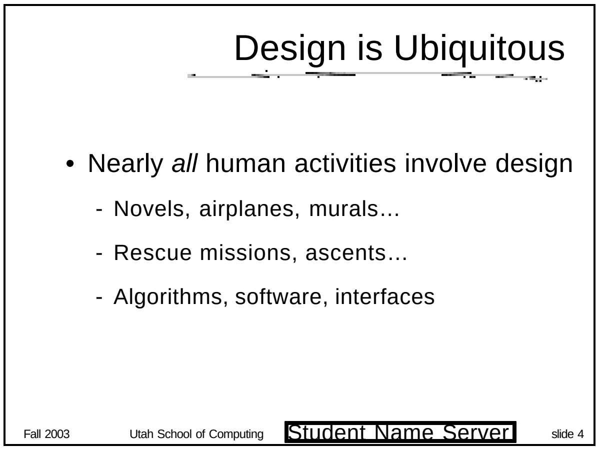
Thesis

- HCI intrinsically involves *design*
  - *"Design an interface to ..."*
- What does this observation entail?



Whereas…

- Design is as old as creativity
- Intensively studied subject
- Much is known
- Let's tap this understanding and experience!



Design is Ubiquitous

- Nearly *all* human activities involve design
  - Novels, airplanes, murals…
  - Rescue missions, ascents…
  - Algorithms, software, interfaces

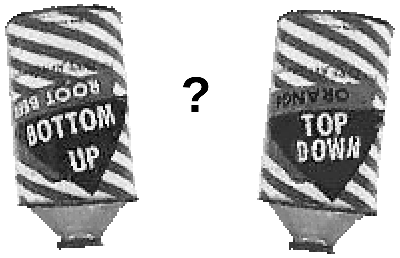
Design Approaches

- Top down
  - Mechanical linkages, compilers, software system
  - Airplane, eg: mission, configuration, weight
  - Recursive refinement technique
- Particular as an instance of General
  - Parametric design

Design Approaches (2)

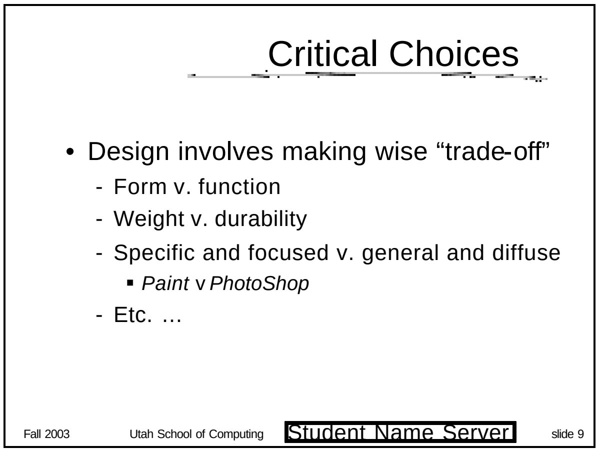
- Bottom up
  - Prototype, gain experience
  - Abstract principles
  - Scale up; begin slow
- Infer *General* from *Particular*
  - *Linguistics*

Note: Bottom_Up **?** Top_Down



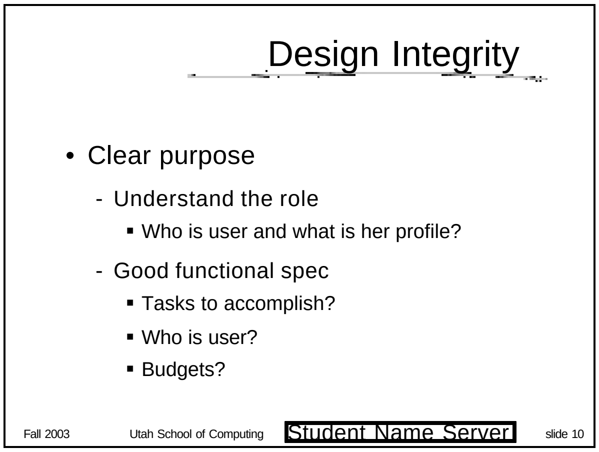
Design Challenges

- Economics
  - Make it good and cheap
  - "*Better, faster, cheaper*"
- Constraints
  - Not design without constraints



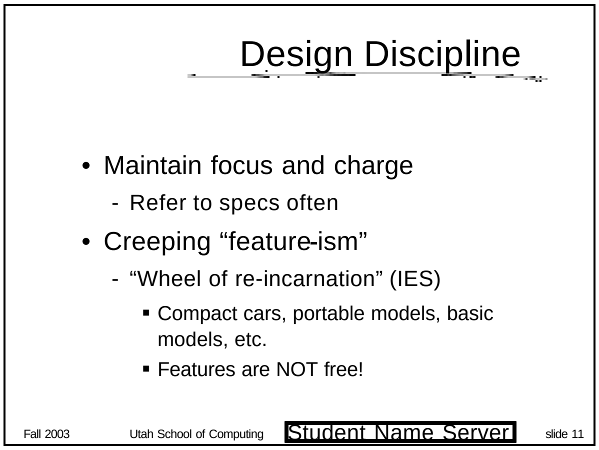
Critical Choices

- Design involves making wise “trade-off”
  - Form v. function
  - Weight v. durability
  - Specific and focused v. general and diffuse
    - *Paint* v *PhotoShop*
  - Etc. …



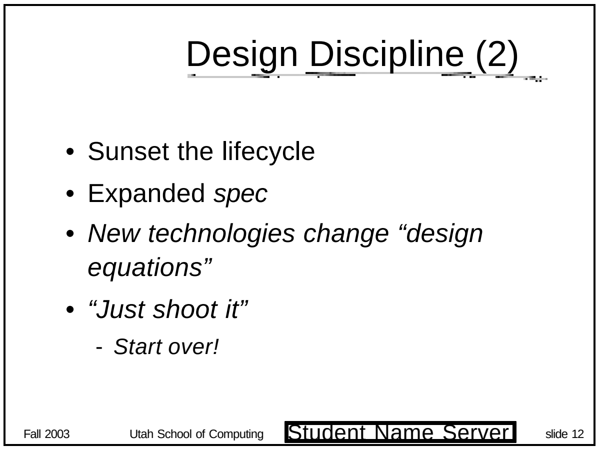
Design Integrity

- Clear purpose
  - Understand the role
    - Who is user and what is her profile?
  - Good functional spec
    - Tasks to accomplish?
    - Who is user?
    - Budgets?



Design Discipline

- Maintain focus and charge
  - Refer to specs often
- Creeping “feature-ism”
  - “Wheel of re-incarnation” (IES)
    - Compact cars, portable models, basic models, etc.
    - Features are NOT free!



Design Discipline (2)

- Sunset the lifecycle
- Expanded *spec*
- *New technologies change “design equations”*
- *“Just shoot it”*
  - *Start over!*

Design Phases/Stages

- Conceptual
  - Show that idea can work
- Preliminary
  - Sufficient to understand, cost, etc
- Detail
  - The “whole enchilada”
  - Adequate for contracting

“Design Intent”

- Why did the designer do this?
- What is the function of this component?
- What was the designer thinking?
- What are the implications if this is modified?

Design History

- Better at design than documentation
- Not sensitive to capturing the past
- Important for the future of a product
- Need better tools
- Record the history as well as final result!

Documentation

- Should not be a post-process
- Capture at time of creation
- Hard problem, actually
  - Who should do it?
  - How should it be accomplished?
- Expensive
  - Not always part of deliverable!

Design Conventions

- Use standards for components
- Use standards for style
- Don’t re-invent terms, tech, tools, etc.
- Make it as straightforward as possible for others who work with you

Variant Design

- Most designs are not really new from the bottom up!
- *Redesign* is far more common as an activity than design, actually
- Make use of the past
- Use templates, components, previous knowledge, catalogs, etc.

Lifecycle Design

- Consider the entire life of a product
  - Cradle to grave (incl disposal)
  - Look at lifecycle cost!
  - Who will maintain?
  - How long will product live?
  - What tools are appropriate?
  - Situations change!

Design for Change

- The only sure thing about a design is that its requirements will evolve and may change dramatically
- Build it flexibly, modularly, clearly wrt to intent, etc

Design Spiral

- Iterate repeatedly
- Budget for interaction
- Throw away early attempts as learning exercises
  - Steve Coons “I know what to throw out.”

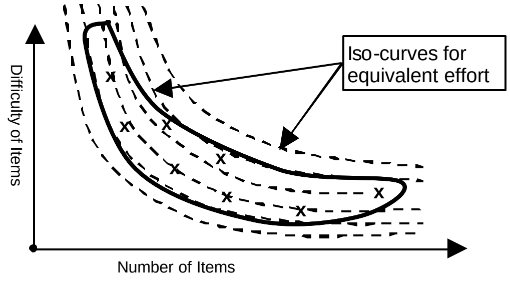
“*ilities*” of Design

- Maintainability
- Portability
- Readability
- Flexibility
- Testability
- Etc, etc….

Complexity “Banana”

- Complexity space often is shaped like a banana:
  - Many simple instances
  - Few complicated instances

Banana Envelope



Design “Reuse”

- Try to make the parts re-usable for other things or future renovations
- Use existing parts if available and of adequate quality

Design is “team sport”

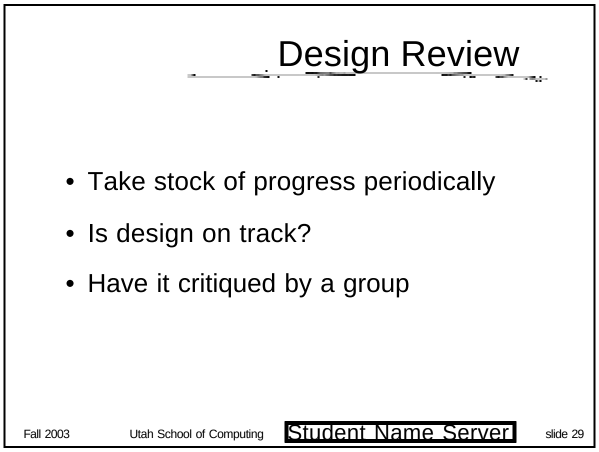
- Most designs involve more than one
- Interfaces are critical, not just components
- Communications, small granularity exchanges, important
- Negotiation, compromise part of deal

Design Views

- Components may serve different functions
  - Different designers see different views
    - Pockets v. Ribs
    - Manufacture v. Structures

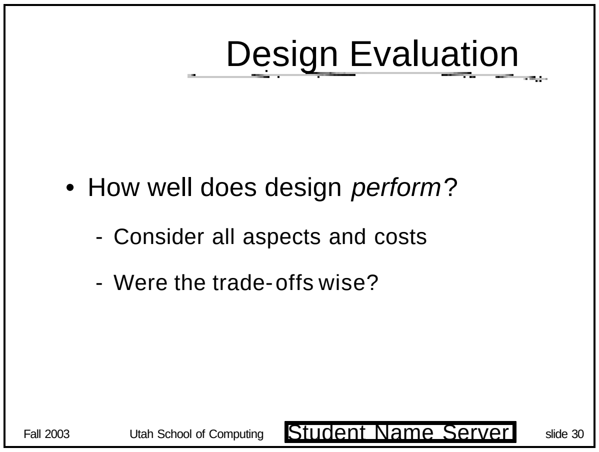
Testing and Validation

- Important stuff!
- Expensive phase
- Underdone activity
  - Alpha testing
  - Beta testing



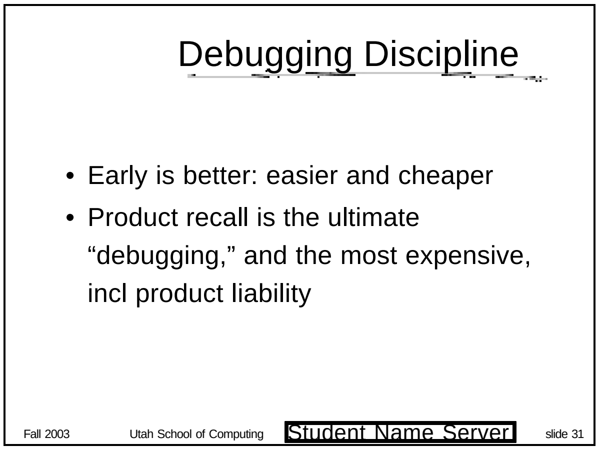
Design Review

- Take stock of progress periodically
- Is design on track?
- Have it critiqued by a group



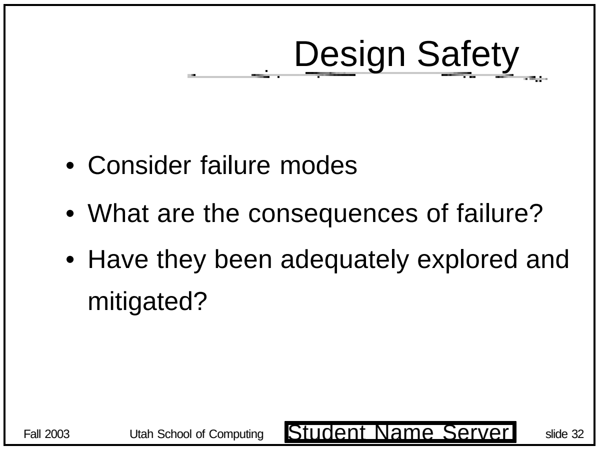
Design Evaluation

- How well does design *perform*?
  - Consider all aspects and costs
  - Were the trade-offs wise?



Debugging Discipline

- Early is better: easier and cheaper
- Product recall is the ultimate "debugging," and the most expensive, incl product liability



Design Safety

- Consider failure modes
- What are the consequences of failure?
- Have they been adequately explored and mitigated?

Design *is* a Creative Process

- Respect its needs
  - Time and patience
  - Concentration, protracted focus
  - Freedom to explore new ways
    - Liberation from past
  - Individual encouragement and support
    - Most ideas are not “keepers”

Consider Multiple Solutions

- Competing prototypes
  - Learn more about merits and liabilities
- Gain experience
- “American way (free market)…”
  - Can help evoke “best effort”

Msg: Recognize Design Activity

- Encourage good design practice
- Nurture good design through better understanding of its nature
  - Establish and protect a conducive environment
- You *are* designers! Do it well!

Utah School of
Computing

The End

Design Methodology