WRAP UP

What you can now do:

• *We have addressed how to design the content of quality interaction, and a process by which usability can be ensured in user interaction.*

• Apply interaction design *guidelines*

• Use an iterative, evaluation-centered *star life cycle* for user interaction development

• Participate in *systems analysis*, including user, needs, task, and functional analyses

• Perform conceptual and detailed *design*

• Establish *usability specifications*

• Build *rapid prototypes*

• Perform *formative usability evaluation*

• Iteratively refine an interaction design

• Know how to get started with these new ideas
WRAP UP

Your biggest challenge may be:

• Not technical!

• **Selling this to management**

  * By necessity, the interaction development process has changed from linear to iterative, which in turn changes at least:

    - Control
    - Scheduling
    - Organizational roles
    - Territoriality
    - Project management
    - Communication
    - Skills
    - Test facilities
    - Tools
• What we've presented is the basis for controllability, accountability, and quantitative methods that are so important (rightfully) to management
WRAP UP

Most important thing: Get management buy-in/education

Selling these techniques to management

• They may not be aware that there is a problem

• They may view these techniques as a solution to a non-existent problem

"The product is selling well"
"We're getting lots of interested inquiries"
"Users don't complain about it"

• "We've never done it this way before."

• Seek out corporate mission statement and show how usability engineering supports

• Remember the "personware" factor

• Successful interaction designs are being developed using these techniques, because they've been shown to work!

• Resources needed: Minimum 15% of entire development effort!

• "You have to keep running just to stay in the same place!"
Red Queen's race, in Alice
COST JUSTIFICATION OF USABILITY

You may be saying to yourself: All this sounds good, but...

Can we afford to include usability engineering in our system development process?

• Answer: Usability engineering does not add overall cost, for two reasons

  * Costs are added only to a limited part of total development process

  * Usability saves many other costs

  First reason:

  • Added costs are confined

    * Reality: Interaction development process must be iterative — cannot get it right the first time

    * But interaction development is small part of overall system development

    E.g., Using RP, not UX "software"

    * Rest of development — user interface software and other application software — is not necessarily iterative
COST JUSTIFICATION OF USABILITY

Second reason:

• Poor usability is costly; good usability is all about saving cost

Usability is about good business, not just about “being nice”

* Costs of hardware and software vs. costs of "personware"

* Costs of development vs. costs of operation
  - Development costs are mostly one time; operational costs accrue for years
  - Cost/benefit scope must be broad enough to include usage, as well as training, help desk support, etc.

Scope problem: one group pays for development cost and another group gets benefits.
COST JUSTIFICATION OF USABILITY

• What costs the most:

  * Finding a design problem early or late in development process?

    Early changes cost 1/4 of changes after installation [Mantei & Teorey, 1988]
    "powers of 10" rule for cost of changes: very early phase = $1; same change at phase 2 = $10; at phase 3 = $100; at phase 4 = $1000.

  * User-based tasks that are quick, efficient, and accurate, or error-prone tasks that take more time?

  * Confused users, or confident and competent users?

    Good usability gives right answers to each of these questions--added value to entire product, benefits received every time it's used
COST JUSTIFICATION OF USABILITY

• Development savings from usability in process
  * High software maintenance costs — trying to get it right after release
  * Implementation costs

  *Bell labs example: saving significant development costs by discovering unneeded functionality*

• Usage savings; even more significant if users are your employees!
  * Save operational productivity costs

  *Esp. Large nbrs of users and repetitive tasks*
  * Save user training costs
  * Save costs of user errors
  * Save costs of database errors
  * Save costs of help desk and user support operations
  * Save intangible costs of employee dissatisfaction

  *Point: Not more resources to ensure usability, but different resources with different distribution during life cycle*
COST JUSTIFICATION: A SIMPLE EXAMPLE

• For a large distributed system:
  Users: 75,000
  Average transactions/user a day: 20
  Transactions/day: 1,500,000
  User time per transaction: 5 - 20 minutes
  Average time saved per transaction, due to improved usability: 30 seconds
  Average fully-loaded hourly rate: $25.00

• Saved per year

  $71,875,000.00

• Other savings: user training, help desk

• Regardless of what usability engineering cost for this product, payback is enormous
COST JUSTIFICATION OF USABILITY

• But won't it be nice when we no longer have to justify "costs" of usability?
"costs" in "", to re-inforce that costs really are associated with NOT including usability in process

• When have you heard anyone ask: Can we afford costs of designing data structures, implementing algorithms, doing quality assurance, etc....?!
These are all considered essential to building an interactive system. BUT we also have to build a UI, and it needs to be competitive/usable. Should we even separate usability for a cost/benefit analysis?!

Remember: anyone can build a UI... (not anyone can create code, do QA, etc...), but...
GETTING STARTED

Some ideas for selling these techniques to management:

• *Start small*

  * Try the process on a small part of a project
  * Try a few usability specifications
  * Set up a small usability lab somewhere, anywhere — and use it
  * Develop at least a minimal customized style guide

• Tell management exactly what you intend to try and hope to accomplish, and within what time frame

• Expect some rough spots in initial stages
GETTING STARTED

Some more ideas for selling these techniques:

• Get appropriate resources lined up
  * Get buy-in from management
  * Get at least one person with appropriate skills on the user interface development team, and give them a title, responsibility, and authority
  * Give appropriate training to team members
  * Get commitment from team members to try these new techniques
  * Find someone you can apprentice with
  * Get consulting help when needed, especially during start-up

   Both in-house and outside
GETTING STARTED

Some more ideas for selling these techniques:

• Professional preparation
  
  * Go to appropriate conferences — e.g., Computer-Human Interaction (CHI); Human Factors and Ergonomics Society (HFES); User Interface Software and Technology (UIST); National Institute of Standards & Technology (NIST)
  
  * Subscribe to HCI publications
  
  * Join Usability Professionals' Association (UPA)
  
  * Join Special Interest Group on CHI (SIGCHI) — local and/or national
  
  * Start a "brown bag" user interface lunch bunch

• Try the process all the way through once

• Generate a failure story
  
  What will it take to let you try all this? Failure!

• Better: Generate a success story
  
  E.g., videoclips for "before and after"
PARTING WORDS

• Encourage focus on the process, rather than just the product
  * Make a customized process guide
  * Include process requirements/strategies in RFPs or in responses to RFPs

• More and more sales and contract awards are being strongly influenced by a demonstrated sound approach to user interaction development — ensuring usability is critical

Anyone can build a user interface, but can they ensure that it is usable?
You’re here because you’re aware of problems

• Be wary of usability “by decree” rather than “by practice”

• Characteristics needed by user interaction/interface developers:
  * Dedication — to the cause of quality interfaces
  * Daring — to do things differently

Art and science in user interface development...