Introduction to Human-Computer Interaction

CS 3724
Fall 2005

Software Engineering and Usability Life-Cycle


Software Life Cycles

- Typical top-down approach
  - Functional decomposition
  - Systems analysis → requirements
    specification → design → prototyping →
    implementation → testing
  - Starts with formal specifications
  - Basically sequential (may feedback to
    previous step)

Waterfall model

Alternating Waves

- Top-down
  - Abstracting, structuring, analyzing
  - System view
  - Too much top-down can lead to ignoring
tough problems until too late

- Bottom-up
  - Concrete, creative
  - User's view

Build a house example
Spiral model

- Boehm (1988)
- Recognizes the need for iteration
  - "a kind of large cycle moving several times through the whole top-down process, each time broadening the circle" (Hix and Hartson, 1993, p. 97)

- Important point to recognize:
  - Because of unpredictable user behaviors, interaction specification is "inherently iterative" and must be "a self-correcting process" (Hix and Hartson, 1993, p. 97)

Usability and Software Engineering

- Software design
  - High-level of abstraction
  - Algorithms, data structures
  - No coding details
- Software implementation
  - Focus on coding
  - Not concerned with design analysis

Usability and Software Engineering

- Two problems
  - Not a perfect separation between roles
  - Specifications can be incomplete or incorrect
- Need communication between roles
  - Can be very formal (documents)
  - May be informal (meetings)

- Question: What happens if implementer tries to address problems with the specification?
Star model
- Hartson and Hix, 1989
- Start almost anywhere
- Recognizes the interconnected-ness of activities
- Evaluation-centered
  - Evaluate results of one activity before proceeding to the next
- Iterative
  - Need usability specifications and evaluation metrics to decide when to stop iterating

Abstract spiral model

Spiral Unwound
Interaction Development Principles

- Empirical testing
  - Early and continuous
  - Representative users
  - Representative tasks
- Iterative Refinement with cost/benefit analyses
- Management process that
  - Verifies and controls dev life cycle
  - Assigns accountability