CS5340
Human-Computer Interaction

January 31, 2013

Today's Class

- Administrivia
- User-centered Design
- Establishing Requirements
- Task Analysis
- Paper Presentations: Social Health Applications
  - Yang Huang
  - Yucheng Huang
  - Cherry Kedia

Break Midway

Administrivia

Assignments

- The best assignments will
  - Demonstrate that you are incorporating knowledge from the class
    - Textbook
    - Research Papers
  - So, for T2/I2
    - concepts from last week's lecture: exemplified in your field notes, etc.
    - Research papers, e.g…
      - how you tried to defamiliarize a familiar space
      - lessons learned from interviewing approach in Wyche et al.

Homework: T2/12

- You should be well on your way to at least securing a location to visit
- Individually-graded
  - 1 unique report per team member
- Field notes should show that you've done readings, paid attention to lecture
  - Note new chapters on Piazza (Lofland, Emerson)

Homework: T2/12

Due 2/7

- Goal: gain first-hand knowledge about your problem space
- As a team, choose a setting to observe
  - You must be resourceful (student groups, community organizations, etc.)
  - Must help you gain insight into topic area
- **Each person**: conduct 2.5 hours of observation
  - Of your target user group

Homework: T2/12

- Tasks
  - Choose location
  - Prior to heading out, talk as a team about what you expect to see
    - Surface biases
    - Identify questions that you hope to answer

Homework: T2/12

- Tasks cont.
  - **Independently,** conduct 2.5 hours of observation
    - **Do NOT** bring phone, computer, tablet etc.
    - Upon arrival, check in with someone in charge to let them know what you're doing
    - If they are not comfortable with you being there, **leave**
    - Make jottings (notes) about what you see
      - Activities, environment, interactions
    - Spend at least 30 minutes interviewing 2-3 people

Homework: T2/12

- Tasks cont.
  - Afterwards, create full field notes
    - Quick notes → prose, quotes
- **Each team member turns in own report**:
  - Jottings
  - Summary of why you chose this setting & questions you hoped to answer
  - Interview questions
  - Full field notes (w/quotes)
  - Implications for design (5 bullet points)

Homework: T2/12

- Carefully read T2 instructions on course website
- Keep an open mind!

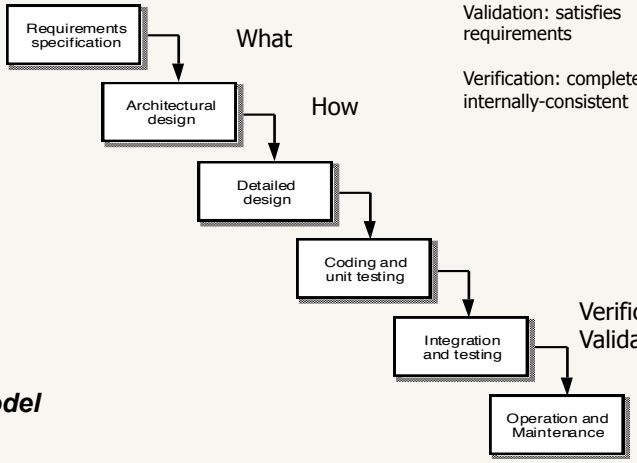
Paper Presentations

- Be sure to post slides to your website, by 5pm on the day of your presentation.

User-Centered Design

The Software Lifecycle

Customer vs. user



What

How

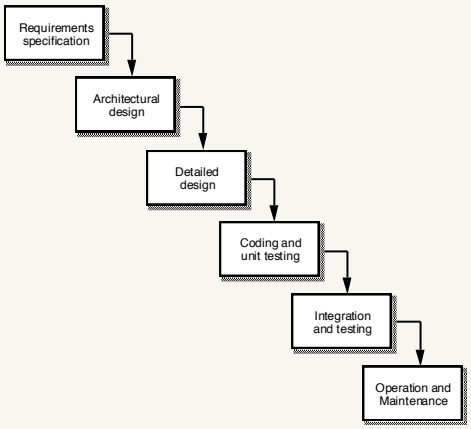
Validation: satisfies requirements

Verification: complete and internally-consistent

Verification v. Validation

The waterfall model

Why doesn't this work for UIs?

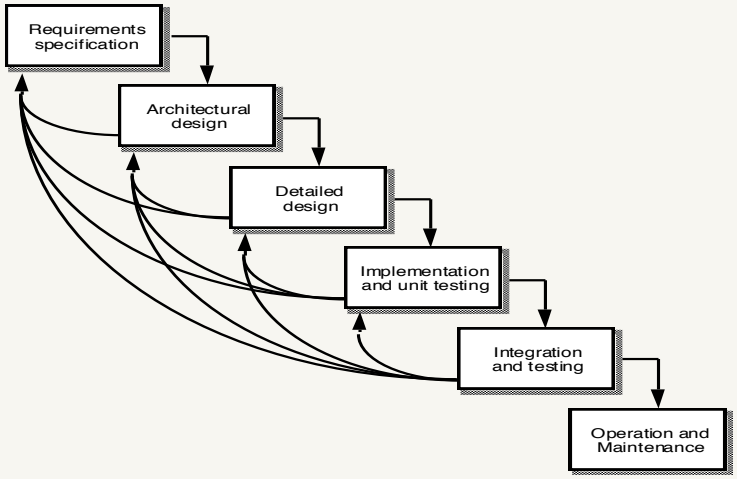


People are insanely complicated – models cannot fully predict how they will use systems (especially early).

Cannot determine all requirements from the start

(which results in 50% designer's time spent on code for UI)

Lifecycle for UIs



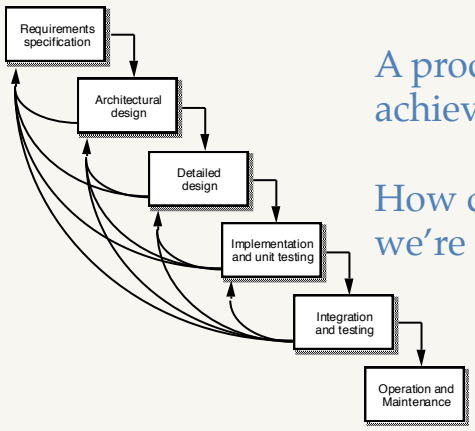
User-Centered Design

- Needs, wants, context of user given primary focus at each stage of the design & development process.
- Repeatedly try out ideas. See how users respond.
  - Involve representative users in all stages of the development process.
  - Minimize the cost of and commitment to prototypes.
  - Users often can't tell you which alternative is "better" – have to test, measure & observe.

User-Centered Design

- Some approaches
  - Methodologies
    - Usability Engineering
    - Iterative Design
    - Contextual Design
    - Participatory Design (later)
  - Methods
    - Qualitative
    - Quantitative
    - Mixed
    - Formative
    - Summative
  - Design Rationales

Usability Engineering



One approach to UCD.

A process by which we achieve system usability.

How do we know when we're there?

Usability Engineering

- Must define usability attributes (multi-dimensional)
- Must define specific measures for each
- Must define "good enough" (goal) levels for each
  - If appropriate, current & ideal levels for each

- Example attributes (measures?)
  - Learnability
  - Efficiency
  - Memorability
  - Low error rate
  - Subjectively pleasing

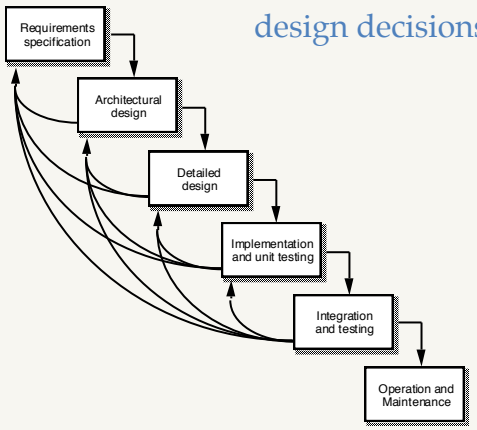
Usability Metrics

- See tables 6.1-6.4

- Limitations?
  - Very specific actions in specific situations
    - Can't account for all scenarios!
  - Doesn't answer whether satisfying metrics yields a sense of usability for user
    - Triangulate to address this

Usability Engineering

How do we remember what we've tried/ learned before and why we made various design decisions?



Design Rationales
+ Decisions made
+ Alternatives investigated
+ Why certain alternatives chosen

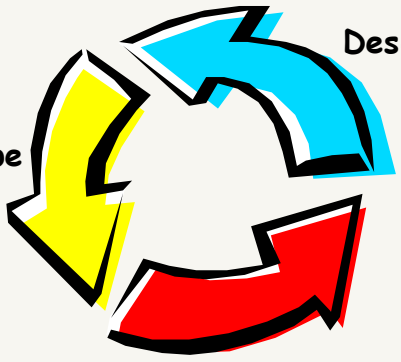
Design Rationales

- Simplest version
  - Documenting major design decisions & rationale
- Your team should be doing this – will help with final report!
- Benefits?
  - Thinking through design trade-offs
  - Communication amongst team
  - Accountability for decisions made
  - Reuse of knowledge across products

Design Rationales

- Process-oriented
  - Captures chronological order of decision making
- Structure-oriented
  - Post-hoc description/re-creation of explored design space
  - Based on criteria, denotes which decisions were favorable/unfavorable
  - Abstracts away from details of process; useful for future design

Iterate!



Ideation
Interaction Design
Beyond GUIs!
User Experience

Paper
Semi-Functional
Functional
UI Toolkits

Observation
Interviews
Focus Groups
Task Analysis

The HCI Development Process

Iterative Design

- Cycling through several designs
- Incrementally improving to reach final product
- Gathering user feedback throughout the process
- Prototyping approaches
  - Throw-away
  - Incremental (one component at a time)
  - Evolutionary (gradual refinement, building)
- How many have done UI development in industry?

Iterative Design

- Beware of ...
  - Early commitment ... Design inertia may make it difficult to recover, even in the face of overwhelming evidence
  - Understand reasons for problems, not just detecting symptoms (and patching)

Establishing Requirements

Requirements…

- …gathering, capture
  - exist "out there", just need to get them
- …elicitation
  - Others know, we just have to pick them up
- But really, they are not that easy to identify
  - Users/customers may not be able to articulate what they want, need

Requirements…

- …engineering
  - Iterative process of negotiation and evolution of req's
- Establishing Requirements
  - Developed from a sound understanding of users' needs
  - Can be justified and related back to the data collected
  - Several data gathering techniques (more later)

What are Requirements?

- A statement about an intended product that specifies
  - What it should do or
  - How it should perform
- Help design team understand as much as possible about users, tasks, context
- Clear, specific, unambiguous
  - To identify when they've been addressed

Requirements

- Functional
  - What the system should do
  - e.g., "game must provide challenging levels for a range of player types"
- Non-functional
  - Constraints on a system and its development
  - e.g., "game must run on multiple platforms"
  - System memory requirements, response time, etc.

Requirements

- Data
  - What kinds of data need to be stored?
  - How will they be stored (e.g. database)?
  - Type, volatility, size, persistence, accuracy

Requirements

- Environmental
  - Context (circumstances) in which the product will operate
  - Physical
    - E.g., dusty? noisy? vibration? light? heat? humidity?
  - Social
    - E.g., sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
  - Organizational
    - E.g., hierarchy, culture, user support, communications structure and infrastructure, availability of training

Requirements

- Users: who are they?
  - Characteristics: ability, background, attitude to computers
  - System use: novice, expert, casual, frequent
    - Novice: step-by-step (prompted), constrained, clear information
    - Expert: flexibility, access/power
    - Frequent: shortcuts
    - Casual/infrequent: clear instructions, e.g. menu path

What are Users' Capabilities?

- Humans vary in many dimensions
  - size of hands → size and positioning of input buttons
  - motor abilities → suitability of certain input and output devices
  - height, if designing a physical kiosk
  - strength
    - a child's toy requires little strength to operate, but greater strength to change batteries
  - disabilities (e.g. sight, hearing, dexterity)

Stakeholders

- Not just the "users"
  - All the people impacted by the introduction of a system
- Consider…
  - People often have conflicting goals
    - Systems assuming cooperation will fail!
  - Must manage these goals, expectations
  - "Symmetry"?
    - work → benefit

Stakeholders: Classes

- Primary
  - End users
- Secondary
  - Receive output or provide input
- Tertiary
  - Directly affected by success or failure
- Facilitating
  - Involved with design, development, maintenance
- Needs often conflict

Priority (not always!)

Stakeholders: Classes

- Primary
  - End users
- Secondary
  - Receive output or provide input
- Tertiary
  - Directly affected by success or failure
- Facilitating
  - Involved with design, development, maintenance

Example:
Electronic Medical Record

Socio-Organizational Issues

- Beyond the interface…
  - How changing social relationships? Culture?
- Power
  - How does technology re-distribute, subvert
- Changing Cultures
- "Invisible worker"

Socio-Organizational Issues

- Free rider problem
  - Examples?
  - How do existing systems address?
- Critical mass
  - # of users whereby the Benefits > Cost
- Evaluating the benefits
  - Not just $$, worker satisfaction, social cohesion etc.

Socio-Organizational Issues

- http://research.microsoft.com/apps/video/default.aspx?id=141399
- How well does this address the "invisible worker" issue?
- What factors would affect this system?
  - Environmental
    - Physical, Social Organizational
  - User
  - Data