Lecture 5-1: Usability Methods II

Design Process *continued*
- Iterative Design: Gould and Lewis (1985)
- User-Centered Design
  - Essential Design Activities: Cohill et al.

- Requirements
- Task Analysis
  - Formal Task Analyses
    - GOMS
    - Other
  - Informal Task Analyses

Exercise

- Oral B CrossAction (white & pink)
- Oral B Advantage
- Reach Max
- Reach Performance (blue & white)

Ergonomic ? *Yes ….*

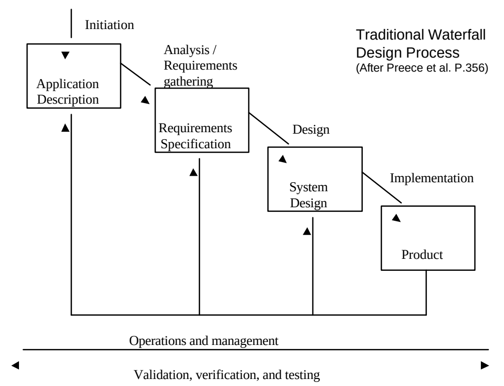
Complete toothbrush user experience ?

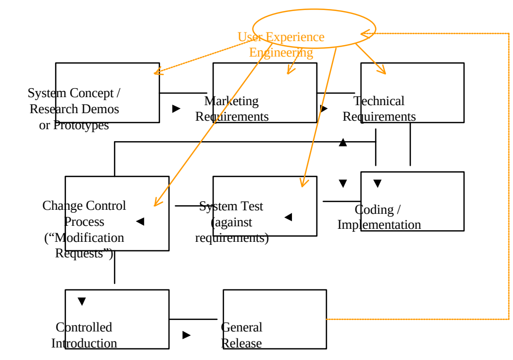
Chap. 16
p. 214

- Petroski, H. (2003) *Small Things Considered: Why There is No Perfect Design*. NY: Knopf.
- " … design must always conform to constraint, must always require choice, and thus always involve compromise. Understanding these elementary facts, we also understand why no design is perfect." (p. 24)

Review

- Models of the design process
  - Important to user experience concern
- Why does design fail?
  - We often know the answers, why aren't they implemented?
  - "Stupid" designers or "stupid" design *process*?
    - Economic, Organizational, social factors
    - No design is perfect - compromises
- Waterfall – traditional design process + modifications
- Gould & Lewis
  - Show that: Idealized design process principles are well known
  - Why aren't they used? Attitudes?





Generalized & Simplified
AT&T / Bell Labs / Lucent Design Process
It's more or less the waterfall model

Iterative Design

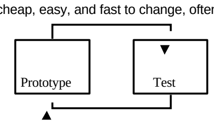
- Gould & Lewis (1985)
  - Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM. 28*(3), p. 300 ff.
- The "key principles"
  - Early focus on users and tasks
  - Empirical measurement
  - Iterative design
- "What designers think"
  - Developers asked to write down design process steps
  - Most developers did not mention the "key principles" (98%)
  - They draw these inferences:
    - Principles are "common sense" but not fully understood
    - There's a difference between ideals and what's actually done

Iterative Design: The Principles (1)

- Early focus on users
  - Must understand who the users are
    - Cognition, behavior, anthropometry, social/attitudes
  - Design team should have *direct contact* with users
    - Interviews, discussions, observations
  - Interviews should be conducted prior to system design
    - Instead of developing a system and presenting it
  - Users should participate in the design
    - Not just "sign off" or "review"
- Empirical Measurement
  - Users should use simulations and prototypes on "real work"
    - This should be done early in the development cycle
  - Performance should be observed, recorded, analyzed
    - Should measure learnability and usability
    - Different from just watching how people use and react to the prototype

Iterative Design: The Principles (2)

- Iterative Design
  - Fix problems found in user testing
    - Cycle of design - test - measure - re-design - etc.
  - A single iteration is not sufficient
  - Use of “testable” behavioral evaluations is extremely important
    - Examples: task completion, number of errors, time taken to complete task, ratings, observations
  - Prototype = Simulation or model of real system without full backend functionality, may likely share no components of actual system
    - cheap, easy, and fast to change, often a “user interface only”



Design Phases

- Initial Design Phase
  - Preliminary specification of the user interface
  - Collect critical information about users
  - Develop testable behavioral goals (Usability goals/criteria)
    - Example: Users must learn basic operations of program in 1/2 hour
  - Organize the work
    - It is best when all aspects of the user interface are integrated in development, e.g. Same group does software and manuals
- Iterative Phase
  - Early testing of features
  - Cheap and flexible method of prototyping

Reasons the Principles are Undervalued (1)

- Some believe the principles are not worth following
- Confusion over what’s recommended
- User diversity is underestimated
  - Don’t realize how different users may be from the designers
- User diversity is overestimated
  - Testing is useless because people are so variable
  - Empirically, test results with sufficiently large sample are usually not idiosyncratic
- Belief that user do not know what they need
  - Not just a matter of asking
  - Must present idea in ways users can relate, or give prototype
- Institutional isolation of designers from users
  - Designers may find it difficult to get user information

Reasons the Principles are Undervalued (2)

- Belief in the power of reason -- why not design logically?
  - Observation can discover what armchair reflection cannot
  - Pre-existing work methods
- Belief that design guidelines are sufficient
  - Guidelines cannot adapt to choices highly dependent on context and task environment
- Belief that good design means getting it right the first time
  - Not possible in user interface design
- Development process will be lengthened unacceptably
  - User testing can be done before system is defined and simultaneously during process of development
  - Prototyping may stimulate development and progress
  - There is a price, but not doing it may come back in support costs and/or costly changes made late in development or after release of product

Reasons the Principles are Undervalued (3)

- Belief that iteration is just "fine tuning"
- Belief in the power of technology
  - Customers will buy the technology in spite of the user interface
  - But, user interface could become market differentiator

Iterative Design Case Study: IBM OMS

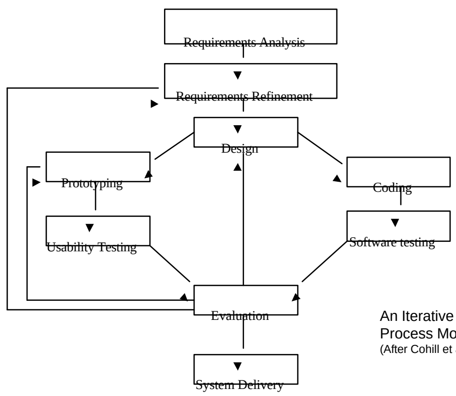
- Olympic Messaging System
  - Voice-mail like system for dropping telephone-delievered audio messages among Olympic athletes
- Process of OMS design (selection):
  - Initial requirements analysis
  - User interface scenarios written
    - Script-like imagined user system phone dialogs
  - Scenarios revised by designers, managers, and users
  - Functionality changed
  - Preliminary user guides written
  - Guides tested on user groups
  - Changed iteratively based upon user feedback
  - Simulations of phone system developed
  - Simulations tested with users
    - Example: uncovered need for "Backup" key

User Centered Design

- User Centered Design:
  - Involvement of user throughout the design process
  - Make user issue central in design process
  - Center on user needs not on technology
  - Test and evaluate with users
  - Iterative design
- User Centered Design can take place in the context of different design process models

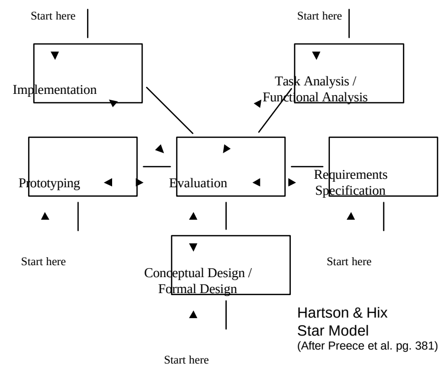
Alternative Design Processes

- Participatory Design ("Scandinavian Approach")
  - Actual users involved at every stage
  - Users sit at design meetings, have roles/jobs in development schedule
- Iterative Design
  - Cohill et al. 1993 manuscript -- reasonable design/prototype process (see diagram)
- Hix and Hartson Star Model
  - Enter at any stage followed by any other stage
  - Conceptual design: requirements, user characteristics, etc.
  - Physical design: actual implementation
- See Preece et al textbook Ch.s 17-18 for more



An Iterative Design
Process Model
(After Cohill et al 1993)

Lecture 5-1 Slide 17



Hartson & Hix
Star Model
(After Preece et al. pg. 381)

Lecture 5-1 Slide 18

Essential Design Activities for HCI (1)

- Essentials for User Centered Design (Cohill et al, manuscript)
- Design Activities:
  - Planning
  - User Needs Analysis
  - Setting Usability Objectives
  - Task Analysis
  - Dialog Design
  - Prototyping
  - Usability Assessment
  - Operational Evaluation
  - Design Process Evaluation

Lecture 5-1 Slide 19

Essential Design Activities for HCI (2)

- Planning
  - Determine sequence and duration of development activities
  - Project Plan document
  - Select design process model
  - Survey users, managers, developers for requirements
- User Needs Analysis
  - Usefulness (Utility) = Relevance or suitability of system functionality to user or task to be performed. Do functions serve a real need for user's job
  - Usability = ease of learning, ease of use, effectiveness, efficiency, and satisfaction
  - At this stage: assess user skill and experience, document user goals and cognitive processes

Lecture 5-1 Slide 20

Essential Design Activities for HCI (3)

- Set Usability Objectives
  - Specify objectives or results users should achieve with system
  - Specify system so that user will achieve goals
  - Specify user attributes (experience, education, training, etc.)
- Task Analysis
  - Study current user work situation to understand requirement of the tasks
- Dialog design
  - Specify the human-computer interaction
  - HCI designer writes user requirements

Essential Design Activities for HCI (3)

- Prototyping
  - Check prototype against requirements
- Usability Assessment
  - Usability audit and/or user testing of prototype or system
  - Iternative prototyping and testing
- Operational evaluation
  - Evaluate usability of system after in use in field
- Design Process Evaluation
  - Conducted after some period following introduction of product

Requirements

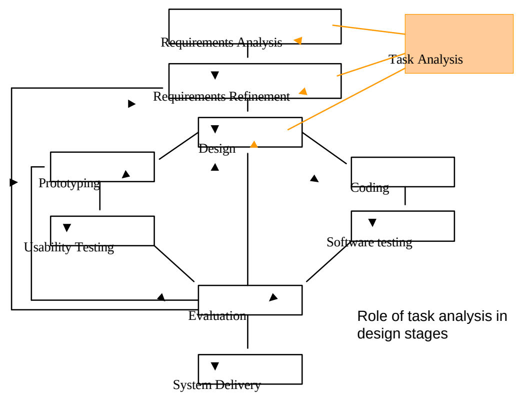
- Functional requirements
  - Details what the system must do
  - Functions and features
  - *Human-computer* system
- Data Requirements
  - Structure of the system and data that must be available
- Usability Requirements
  - Set acceptable level of usability performance and satisfaction (usability objectives)
  - Specified in terms of performance measures call *usability metrics*
    - Time to complete task by specified set of users
    - Errors
    - Time spent reading documentation
    - Etc.

Functional Requirements

- User interface designers should have important role
  - Specification of user interface in separate functional document
  - Or, responsibility for user interface section in system engineering requirements
  - Review and editing of all functional and architectural requirements for impact on usability and user interface
- Usefulness issues may most often surface in marketing requirements
  - User interface could have an authoring role, or interact with marketing
- User Interface Requirements:
  - Textual descriptions, tables
  - Screen mock-ups
  - Flowcharts

Task Analysis

- Formal (quantitative) task analysis
  - GOMS
- Informal ("practical") task analysis
  - Day and Boyce pp. 370-381 (section 6)
- Task analysis
  - Early focus on users and tasks
  - Class of methods for understanding human in human-computer system
  - Evolved from time-and-motion studies in 50s
  - Modern formal task analyses use estimations and analyses from cognitive psychology memory and performance research and theories



Role of task analysis in design stages

Formal Task Analysis

- Highly structured, formal systems for describing user and task(s)
- Hierarchical Task Analysis
  - Logical structure of a task
  - Hierarchical set of plans and operations (subtasks)
- Cognitive Task Analysis
  - Also capture people's knowledge of the task
  - Specify the mental (cognitive) actions as well as the physical
  - Use time estimations and other research and theoretical findings from cognitive psychology
  - GOMS
    - Goals, Operations, Methods, and Selection rules
    - Card, Moran, & Newell (1983). *The Psychology of Human-Computer Interaction.*

GOMS

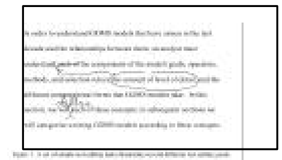
- Early and most prominent version of cognitive task analysis
  - many variations and spin-offs now exist
- Description of Methods (plans) needed to accomplish the user's Goals
- Methods = series of steps, simple tasks ("Operations")
- Selection rules -- needed to choose among alternate methods, if they exist, based upon context
- Representation of users' goals and knowledge = "user model"
- Steps:
  - Fully specify user model
  - Estimate time to complete task form human performance data
  - Convert model to a program which estimates times for different versions of task / system design

Levels of GOMS Models

- GOMS level
  - General methods for accomplishing task(s)
- Unit task level
  - Break into task unit, estimate time it takes to perform (often based upon findings from human performance literature - cognitive psychology)
- Keystroke level
  - Describe and predict time it takes to perform a task by specifying keystrokes needed

GOMS Example

- Kieras (1997) Document editing on Macintosh computer
- Goal = move some text
- Steps → Sub-goals = **cut text** + **paste text**
- Operator = **verify**
- Sub-goals → Methods = selection sub-goals + **retain** operator
- Other goals analyzed:
  - Select word
  - Select arbitrary text
  - Select insertion point
  - Issue command



Goals, Methods, and Operators

Method for goal: **move text**
- Step 1. Accomplish goal: **cut text**
- Step 2. Accomplish goal: **paste text**
- Step 3. Verify correct text moved.
- Step 4. Return with goal accomplished.

Method for goal: **cut text**
- Step 1. Accomplish goal: **select text**
- Step 2. **Retain** that the command is *cut*, and accomplish goal: **issue a command**
- Step 3. Return with goal accomplished.

Method for goal: **paste text**
- Step 1. Accomplish goal: **Select insertion point**
- Step 2. Accomplish goal: Retain that the command is *paste*, and accomplish goal: **use a command**
- Step 3. Return with goal accomplished.

Method for goal: **select insertion point**
- Step 1. Accomplish goals: **cut text**
- Step 2. Accomplish goal: **paste text**
- Step 3. Verify correct text moved.
- Step 4. Return with goal accomplished.

Selection Rules

Select rule set for goal: **select text**
- If text-is-word, then accomplish goal: **select word**
- If text-is-arbitrary, then accomplish goal: **select arbitrary text**
- Return with goal accomplished

From Kieras (1997) pp. 755-756.
Kieras (1997) "A guide to GOMS model usability Evaluation using NGOMSL". In *Handbook of Human-Computer Interaction*. (2nd Ed.) M. G. Helander, T. K. Landauer, & P. V. Pradhu (Eds) (pp. 733-766) Amsterdam: North-Holland.

GOMS Model Human Processor (1)

- The Model Human Processor -- Principles of Operation
  - P0. Recognize-Act Cycle of the Cognitive Processor
    - On each cycle of cognitive processor τ_c : contents of working memory links to long term memory, which modify working memory
  - P1. Variable Perceptual Processor Rate Principle
    - Perceptual processor cycle time τ_p varies inversely with stimulus intensity
  - P2. Encoding Specificity Principle
    - Specific encoding operations performed on what is perceived determine what is stored and what is stored determines what retrieval cues are effective in providing access to what is stored
  - P3. Discrimination Principle
    - The difficulty of memory retrieval is determined by the candidates that exist in the memory, relative to the retrieval clues.

GOMS Model Human Processor (2)

- P4. Variable Cognitive Processor Rate Principle
  - The cognitive processor cycle time τ_c is shorter when greater effort is induced by increased task demands or information loads; it also diminishes with practice.
- P5. Fitts's Law
  - The time T_{pos} to move the hand to a target of size S which lies a distance D away is given by
    $T_{pos} = I_M \log_2 (D/S + .5)$ where I_M = 100(70-120) msec/bit
- P6. Power Law of Practice
  - The time T_n to perform a task on the n^{th} trial follows a power law:
    $T_n = T_1 n^{-\alpha}$ where α = .4 [.2 - .6]
- P7. Uncertainty Principle
  - Decision time T increases with uncertainty about the judgement or decision to be made:
    $T = I_C H$ where H is the information-theoretic entropy of the decision and I_C = 150 [0 - 157] msec / bit ...

Lecture 5-1 Slide 33

GOMS Model Human Processor (3)

- P8. Rationality Principle
  - A person acts so as to attain his goals thru rational action, given the structure of the task and his inputs of information and bounded by limitations on his knowledge and processing ability:
    Goals + Task + Operations + Input + Knowledge + Process-limits → Behavior
- P9. Problem Space Principle
  - The rational activity in which people engage to solve a problem can be described in terms of
    1. **a set of states of knowledge**
    2. **operators for changing one state into another**
    3. **constraints on applying operators**
    4. **control knowledge for deciding which operator to apply next.**

FROM Card, Moran, and Newell (1983)

Lecture 5-1 Slide 34

Simple Example GOMS Calculations (1)

- Simple Reaction Time: Determine the time for symbol appearance until press space bar on standard display terminal

Vision to working memory = 1 cycle of perceptual processor
Connect stimulus with response =1 cycle of cognitive processor
Execute movement = 1 cycle of motor processor
$\tau_p + \tau_c + \tau_m$ = 100 + 70 + 70 = 240 msec.

Lecture 5-1 Slide 35

Simple Example GOMS Calculations (2)

- Two symbols presented sequentially; if second identical to first, then press "YES" button

1st symbol: Vision to working memory = 1 cycle of perceptual processor
Start timing now:
2nd symbol: Vision to working memory = 1 cycle of perceptual
match codes in cog. Processor = 1 cycle of cognitive processor
If match occurs:
Cog. Processor "decides" yes = 1 cycle of cognitive processor
Execute movement = 1 cycle motor processor

$\tau_p + 2\tau_c + \tau_m$ = 100 + 140 + 70 = 310 msec.

Lecture 5-1 Slide 36

Simple Example GOMS Calculations (3)

- Two symbols presented sequentially; if second has same *name* then press "YES" button

2nd symbol: Vision to working memory = 1 cycle of perceptual processor
Convert image to name = 1 cycle of cognitive processor
Match codes in Cog. processor = 1 cycle of cognitive processor
If match occurs:
Cog. Processor "decides" yes = 1 cycle of cognitive processor
Execute movement = 1 cycle motor processor

$\tau_p + 3\tau_c + \tau_m = 100 + 210 + 70 = 380$ msec.

Development of GOMS and Formal Task Analysis Methods

- A GOMS analysis can reveal problems with consistency and unnecessarily complex procedures
- Gray, John, & Atwood (1992) 'Project Ernestine' @ Bellcore
  - GOMS analysis of telephone operator workstation
  - Keystroke savings are worth $$ Millions (due to 'economy of scale')
- Developments
  - Gray et al (1992) extended to multiple, parallel task execution
  - Kieras (1997) extended to learning, perception, and errors
- Payne & Green (1986) Task-action Grammar
- Carroll, Mack & Kellogg (1988) User interface metaphor analysis
  - Grammar like analysis models users' use of analogy (mental model) of existing systems

Practical Task Analysis

- Day & Boyce (1993) Section 6
- Formal methods criticized
  - Too time-consuming for actual development environments
  - Inappropriate for certain stages of design
  - Focuses on individual operations in tasks, thus ignores
    - Overall structure of work and organizational issues
    - Leaning and training
    - Problem solving in error situations
- What really happens?
  - Not enough time
  - Tight schedule constraints and cost limitations
  - Details of task analysis difficult to rationalize as relevant
- "Informal" task analyses are often done, but not written about!

Steps of a Practical Task Analysis (1)

- Focus on a subset of critical tasks
  - Not an exhaustive catalog of tasks and actions
  - Generalize from critical tasks, if possible
  - Understand tasks within context of use (user characteristics, company policies, environment, etc.)

- Step 1: Collection of background information
  - Find out from user / user's company how the system works in the context of the workplace
  - High-level description of systems and procedures

Steps of a Practical Task Analysis (3)

- Step 2: Interviews and observations
  - – Interview users
    - Training/ experience questions
    - Describe their work, list tasks they perform
    - Detailed description of selected tasks (important tasks)
  - – Observe users in their jobs
- Step 3: Data Analysis
  - – Comprehensive task list culled from individual interviews
  - – Priority, frequency, duration of tasks

Steps of a Practical Task Analysis (4)

- Data is summarized:
  - – Hierarchical task diagram
    - Block diagram - high-level view
  - – Hierarchical task description
    - Detailed text description of task and subtasks
  - – Flow-chart
  - – Task parameters table
    - Specify starting conditions or principle inputs to subtasks
- Final step: Make recommendations to development team