Theory

✓ Models of interaction
  - ✓ Affordances, mappings, constraints, types of knowledge, errors, feedback

✓ Design principles
  - ✓ Nine golden rules, responsiveness

⇒ Human cognition and performance

- History and vision of HCI

Practice

✓ Sketching

✓ User observation

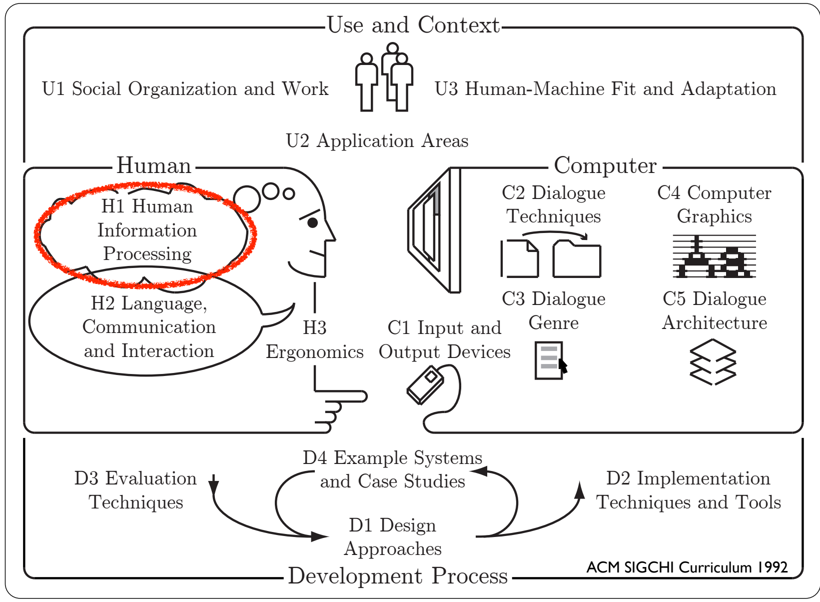
✓ Iterative design

✓ Prototyping

✓ Ideation

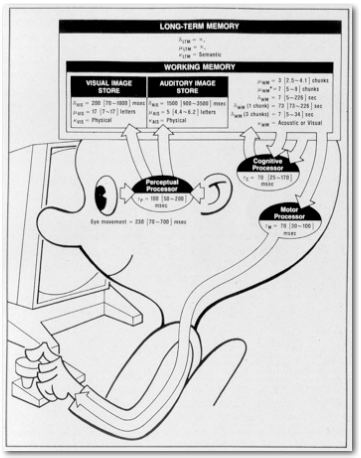
- User study and evaluation

What's Human-Computer Interaction?



Some Cognitive Psychology Basics

- The Model Human Processor
- Perception, Cognition, Motor System
- Fitts' Law



Experiment 1

1cm

4cm

Same with .5cm and 2cm wide strips
Tap for 10s, count taps afterwards

8cm

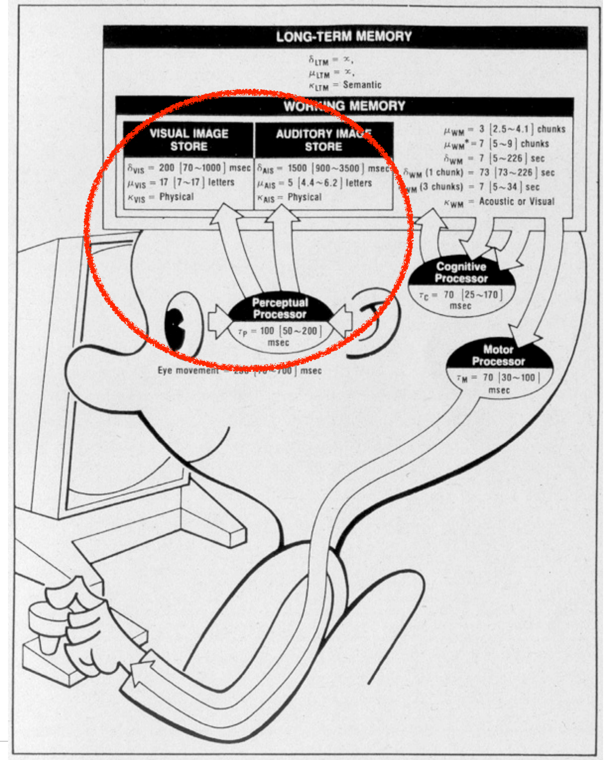
16cm

Model Human Processor

- By Stuart Card, Tom Moran, Alan Newell
  - The Psychology of Human-Computer Interaction, 1983
- Basic model for perception, memory, and motor system
- Goal: estimate execution time, error rates, and training effects for simple input/output events

Model Human Processor

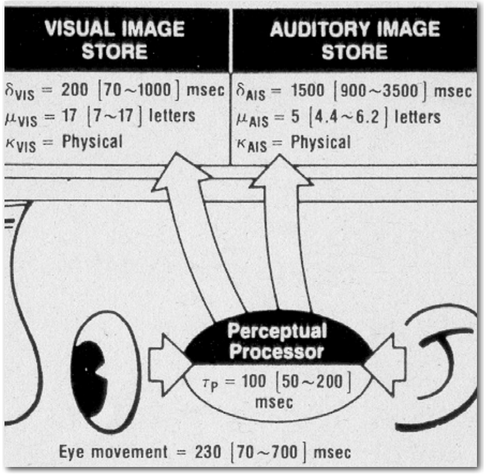
- CMN Model
- 3 processors with associated memory
- SlowMan, MiddleMan, FastWoman



Experiment 2

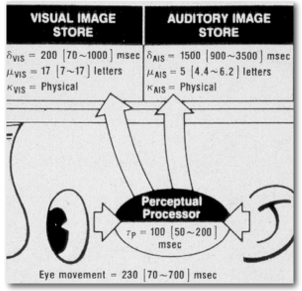
- In pair, one person reads the paragraph handed out.
- Have your friend observe your eye movement.
- How does the eye move while your partner reads?

Perception



- Eye saccades: 230 ms
- Explains reading rates
  - Maximum: 13 characters/saccade ⇒ 652 words/minute

Perceptual Processor



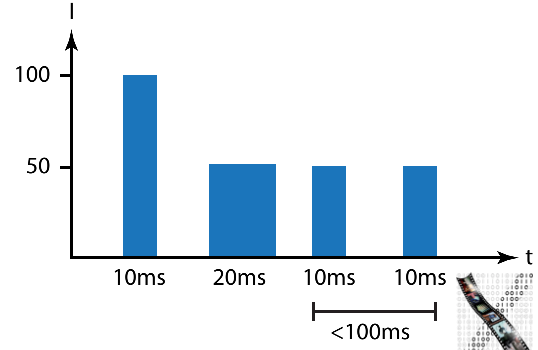
- Stores sensor signals in visual & auditory stores
- Perception time: $\tau_P \approx$ 100 ms
  - Explains Bloch's Law

    $$R = I \times t$$

    R is response

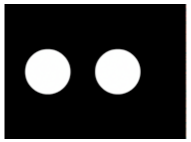
    I is intensity,

    t is exposure time
  - Constant response for t < 100ms



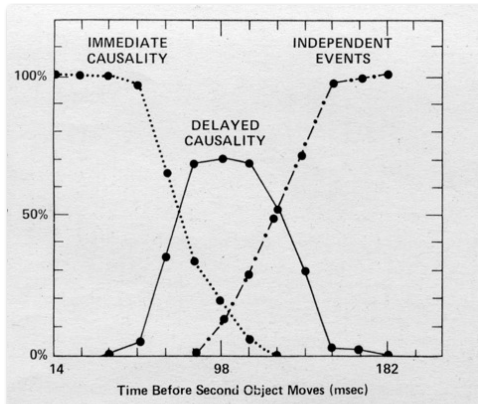
Causality Experiment

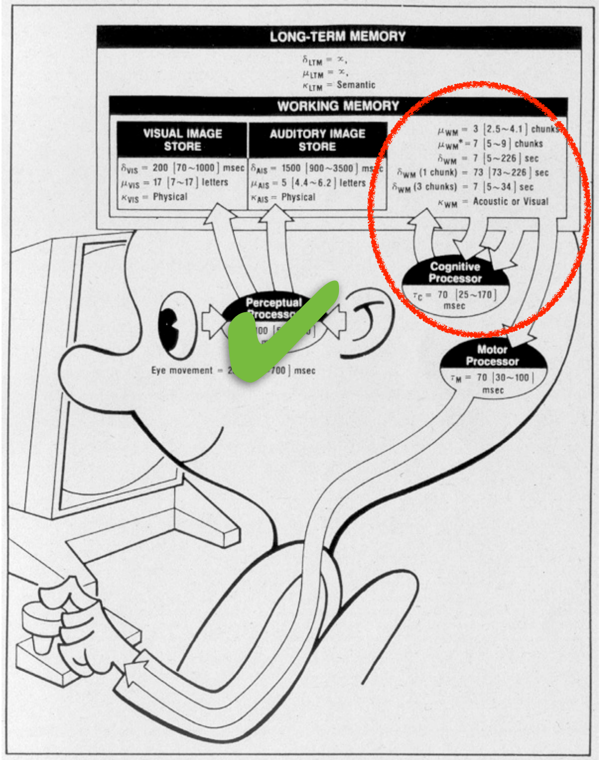
- The animations show two billiard balls colliding, and then the still ball moving after 0, 50, or 100ms delay



Perceptual Processor

- Perception time: $\tau_P \approx$ 100 ms
  - Explains animation rates (10 fps for MiddleMan)
  - Explains max. delay before causality breaks down
  - Shortens with intensity



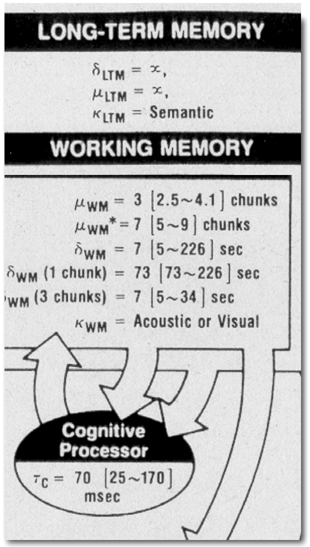


Experiment 3

1. Digit experiment
   1. Choose 5 digits secretly from your sheet, then read them to your neighbor.
   2. Have him count backwards aloud from 50.
   3. Have him answer some other question (like what he had for dinner 3 days ago).
   4. Does he still remember the entire 5-digit sequence correctly?
2. Switch roles, repeat with 9 digits.
3. Finally, switching roles again, read the long sequence of numbers to your neighbor, stopping somewhere suddenly. See how many of the last numbers he can repeat immediately.

Cognitive System

- Chunks depend on user & task
- Working memory:
  - Capacity: μ_{WM} = 7 ± 2 chunks (Miller'56)
  - Half life: $\delta_{1,WM}$ = 73 s (1 chunk)
    $\delta_{3,WM}$ = 7 s (3 chunk)
  - Visual/acoustic encoding

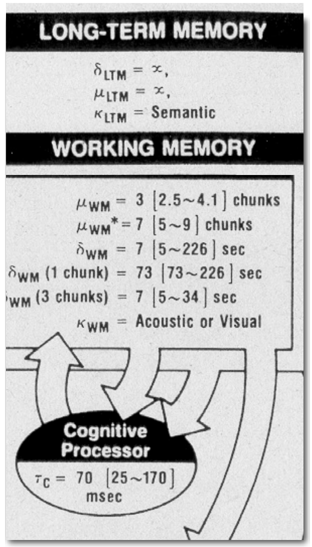


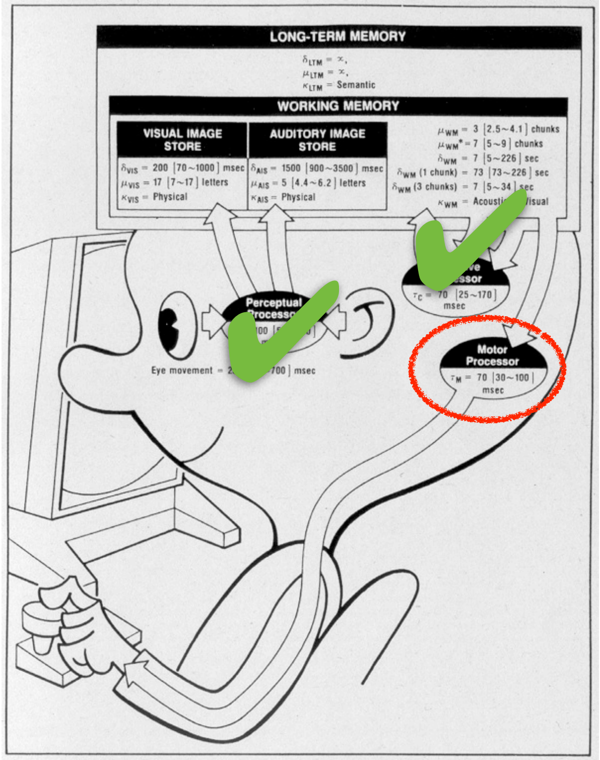
Cognitive System

- Long-term memory:
  - Infinite capacity and half life
  - Semantic encoding (associations)
  - Fast read, slow write

⇒Remembering items maxes out at 7 s/chunk learning speed (1 pass)

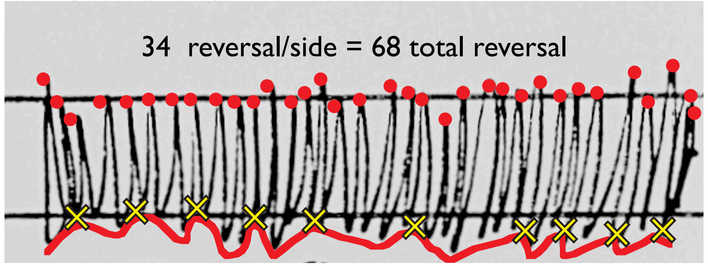
- Cognitive processor:
  - Processing time τ_C = 70 ms





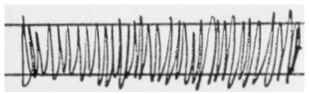
Experiment 4

- Experiment: 5 s of strokes between lines. Try to reach both lines.
- Count number of reversal
  - How many milliseconds per reversal?
- Create a contour of stroke bottoms, count number of corrections
  - How many milliseconds per correction?

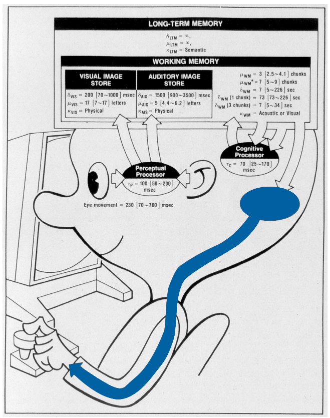


10 corrections/side = 20 total corrections

Motor System

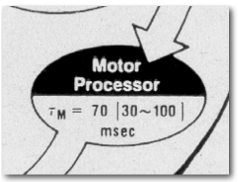


74 ms/reversal
250 ms/correction

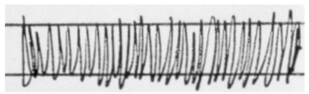


- Motor processor (open loop)
  - τ_M = 70 ms

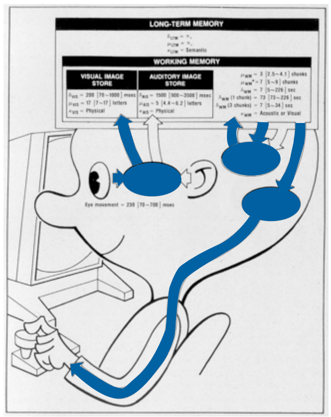
$\Rightarrow$ Average time between each reversal



Motor System



74 ms/reversal
250 ms/correction



- Closed loop:
  - $\tau_P + \tau_C + \tau_M$ = 240 ms

$\Rightarrow$ Average time between each correction

Tapping Task Results

(Experiment 1)

1cm

4cm

Same with .5cm and 2cm wide strips

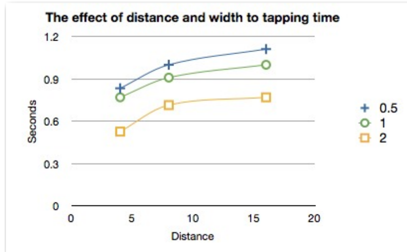
Tap for 10s, count taps afterwards

8cm

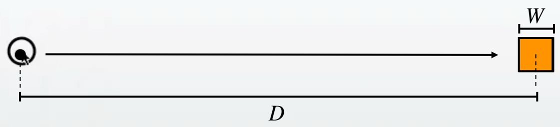
16cm

Tapping Task Results

- Doubling the distance adds roughly a constant to execution time
  - ⇒ indicates logarithmic nature
- Doubling the target width gives about same results as halving the distance
  - ⇒ indicates connection of D/W in formula

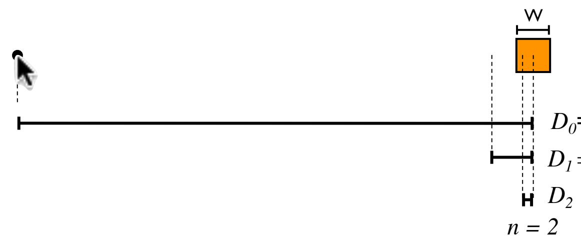


Motor System: Fitts' Law



- Goal: Predict time to press buttons (physical or on-screen) as function of distance and size
- Result (Fitts, 1954): $T_{pos} = I_M \times I_D$
  - T_{pos} time to reach button
  - I_M = 100 ms/bit index of movement, constant
  - $I_D = \log_2(2D / W)$ index of difficulty, in bits

Deriving Fitts' Law from CMN



- D_i: remaining distance to target after i^{th} movements,
- $\epsilon := \frac{D_i}{D_{i-1}} < 1$ relative movement precision, is a constant (CMN)

$\epsilon = 0.07$ from experiment

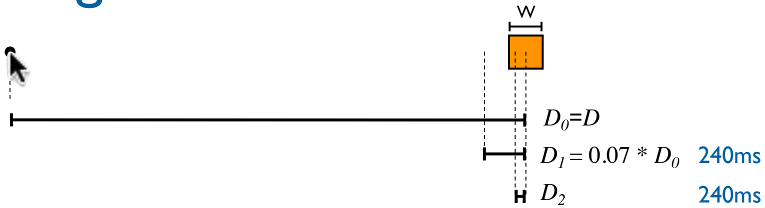
$$D_1 = \epsilon * D_0 \quad = \epsilon \ * D$$
$$D_2 = \epsilon * D_2 \quad = \epsilon^2 * D$$
$$\dots$$
$$D_n = \epsilon * D_{n-1} = \epsilon^n * D \leq \frac{W}{2}$$

(hand reaches target after n movements)

Deriving Fitts' Law from CMN

$\epsilon = 0.07$

$\epsilon^n * D \leq \frac{W}{2}$



- Solve for n

$$\begin{aligned} \epsilon^n * D &\leq \frac{W}{2} \\ \epsilon^n &\leq \frac{W}{2D} \\ n &\geq \log_\epsilon \frac{W}{2D} \\ n &\geq \frac{\log_2 \frac{W}{2D}}{\log_2 \epsilon} \\ n &\geq \frac{-\log_2 \frac{2D}{W}}{\log_2 \epsilon} \end{aligned}$$

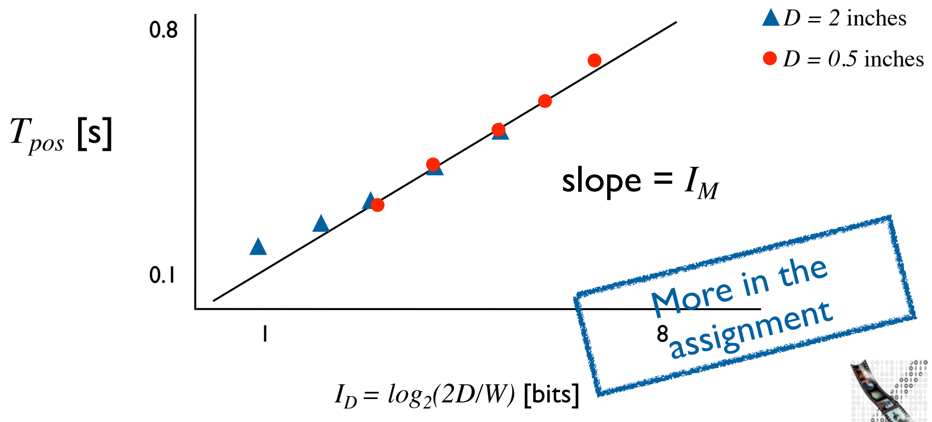
(log for base < 1 turns inequality sign)

- Positioning for n times

$$\begin{aligned} T_{pos} &= (t_p + t_c + t_m) * n \\ &= 240 * n \\ &\geq 240 * \frac{-\log_2 \frac{2D}{W}}{\log_2 \epsilon} \\ &\geq \frac{-240}{\log_2 \epsilon} * \log_2 \frac{2D}{W} \\ &\geq \frac{-240}{\log_2 (0.07)} * \log_2 \frac{2D}{W} \\ &\approx 100ms/bit * \log_2 \frac{2D}{W} \\ T_{pos} &= I_M * I_D \end{aligned}$$

Visualizing Fitts' Law

Experiment: fixed distance D, varying width W



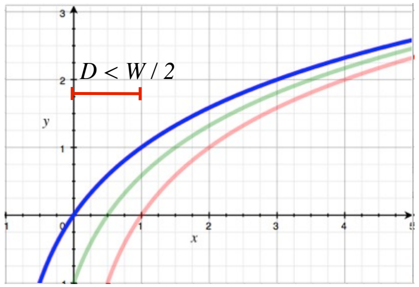
Improvements

- Welford's Formulation, 1968:
  - $T_{pos} = I_M \cdot \log_2 \left(\frac{D}{W} + \frac{1}{2} \right)$
- Shannon's Formulation, ISO, 80's:
  - $T_{pos} = a + b \cdot \log_2 \left(\frac{D}{W} + 1 \right)$

a, b depend on device, determine experimentally

Use a = 0 ms, b = I_M = 100 ms for quick and dirty estimates

Improved curve fit, no negative times for infinite-size targets

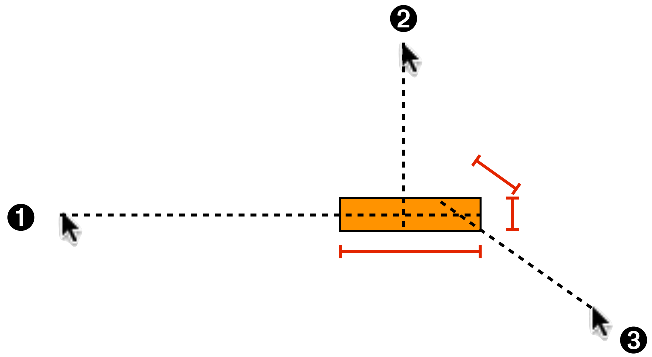


$T_{pos} = I_M \cdot \log_2 \left(\frac{2D}{W} \right)$

$T_{pos} = I_M \cdot \log_2 \left(\frac{D}{W} + \frac{1}{2} \right)$

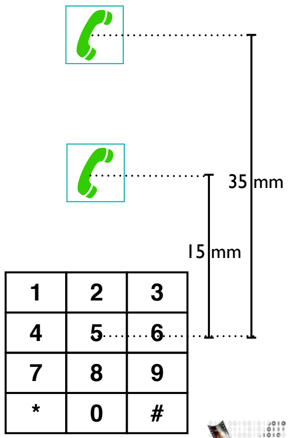
$T_{pos} = a + b \cdot \log_2 \left(\frac{D}{W} + 1 \right)$

Target Width



Exercise: Mobile Phone

- How much faster does calling become by moving the "call" button from 35 mm distance to 15 mm distance, measured from the middle of the keypad? The size of the call button is 10 x 10 mm
- Welford's Law: $T_{pos} = I_M \cdot \log_2\left(\frac{D}{W} + \frac{1}{2}\right)$
- Use I_M=100 ms/bit



Solution

$$T_{pos1} = I_M \cdot \log_2\left(\frac{D_1}{W} + \frac{1}{2}\right)$$
$$T_{pos2} = I_M \cdot \log_2\left(\frac{D_2}{W} + \frac{1}{2}\right)$$

$$\begin{aligned} T_{pos1} - T_{pos2} &= I_M \cdot \left(\log_2\left(\frac{D_1}{W} + \frac{1}{2}\right) - \log_2\left(\frac{D_2}{W} + \frac{1}{2}\right)\right) \\ &= 100\,\frac{\text{ms}}{\text{bit}} \cdot \left(\log_2\left(\frac{35}{10} + \frac{1}{2}\right) - \log_2\left(\frac{15}{10} + \frac{1}{2}\right)\right)\ \text{bit} \\ &= 100\,\text{ms} \cdot (\log_2 4 - \log_2 2) \\ &= 100\,\text{ms} \cdot (2 - 1) \\ &= 100\,\text{ms} \end{aligned}$$

⇒ Moving the call button speeds up each call by an average of about 100ms.

Theory

✓ Models of interaction
  - ✓ Affordances, mappings, constraints, types of knowledge, errors

✓ Design principles

⇒ Human cognition and performance

- History and vision of HCI

Practice

✓ Sketching

✓ User observation

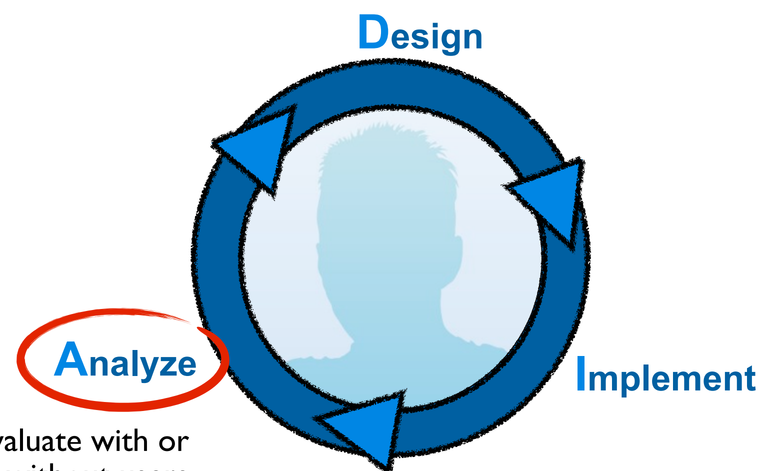
✓ Iterative design

✓ Prototyping

✓ Ideation

⇒ User study and evaluation

When to Evaluate



Evaluate with or without users

Why Evaluate?

- To ensure that system matches user needs
- Necessary even if design was already user-centered (interviews, …)!
- Evaluation should happen throughout the entire software development process
  - Early designs are more often evaluated by design team, analytically and informally
  - Later implementations are more often evaluated by users, experimentally and formally

Why Evaluate?

- To judge system features
  - Does it facilitate users' tasks?
  - Does it offer the right features, easy to reach, and presented as expected?
- To judge effects on users
  - How easy to learn and use is the system?
  - How do users feel about the system?
  - Are there areas that overload users?
- To discover specific problems
  - Do unexpected/confusing situations come up?

Where to Evaluate: Lab

- With or without users

\+ Equipment (A/V, see-through mirrors, special computers), no disruptions, quiet

– Natural environment missing (shelves, wall calendar, …); unnatural situation (relevance?)

- Only place possible if real use dangerous, remote (ISS), or controlled situation needed

Where to Evaluate: In The Field

- Studies in the users' natural environment

\+ More realistic (also *because* of disruptions)

\+ Situations and behavior more natural

\+ Better suited to long-term studies

– Noise, task interruptions

– Will still feel like a test situation

Participatory Design

- Involve user as part of design team throughout entire software process
- Originated in Scandinavia where it is the law for certain products
- Techniques for team communication
  - Brainstorming, storyboarding, workshops, interviews, role plays, paper prototypes
- Problems
  - High effort, conflicts with client hierarchies, user conversion

Evaluation Techniques

Evaluating Without Users

E1 Literature Review
E2 Cognitive Walkthrough
E3 Heuristic Evaluation
E4 Model-Based Evaluation (GOMS,...)

Evaluating With Users

Qualitative

E5 Model Extraction
E6 Silent Observation
E7 Think Aloud
E8 Constructive Interaction
E9 Retrospective Testing

Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

E1: Literature Review

- Many research results about user interface design have been published
- Idea: Search literature for evidence for (or against) aspects of your design
- Saves own experiments
- Results only carry over reliably if context (users, assumptions) is very similar

E2: Cognitive Walkthrough

- Without users
- Expert = designer or cognitive psychologist
- Goal: Judge learnability and ease of use
- Step through each task and ask
  - How does interaction influence user?
  - What cognitive processes will she need?
  - What problems could learning/doing this step have?
- Does system help user to get from goals to intentions and actions?
- Requires interface description, task description, and user profile

E2: Cognitive Walkthrough

- What to do:
  - Choose task—describe goals—determine actions
  - Analyze this decision process using above questions
- Question forms capture psychological knowledge and guide the tester
- Analytical method for early design or existing systems
- Takes time

E3: Heuristic Evaluation

✓ . . .
✓ . . .
✓ . . .

- Variant of Cognitive Walkthrough
- Choose usability heuristics
  - General guidelines, e.g., Nine Golden Rules
- Step through tasks and check whether guidelines are followed

  + Quick and cheap

  – Subjective

    Better done by several independent designers

E4: Model-Based Evaluation

- Some models exist that offer a framework for design and evaluation
- Examples:
  - Information efficiency
  - GOMS, KLM
  - Design Rationale (History of design decisions with reasons and alternatives)
  - Design Patterns

Evaluation Techniques

Evaluating Without Users

E1 Literature Review
E2 Cognitive Walkthrough
E3 Heuristic Evaluation
E4 Model-Based Evaluation (GOMS,...)

Evaluating With Users

Qualitative

E5 Model Extraction
E6 Silent Observation
E7 Think Aloud
E8 Constructive Interaction
E9 Retrospective Testing

Quantitative

E10 Controlled Experiments

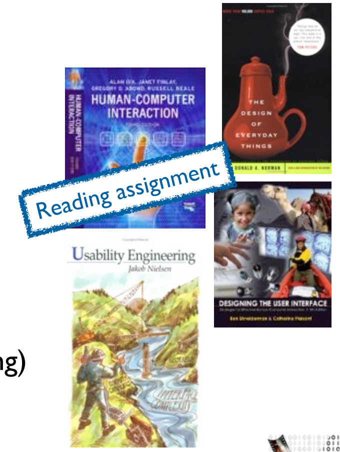
\+ Interviews, questionnaires,...

HCI Literature Sources

- Conferences
  - CHI (main), UIST (technical), DIS (design), INTERACT (Euro-Asian), CSCW (group work), Ubicomp, ACM Multimedia
- Journals
  - TOCHI (archival), Interactions (magazine), PUC (Ubicomp)
- Online
  - ACM Digital Library, hcibib.org

HCI Literature Sources: Books

- Norman '02: Design of Everyday Things
  - Affordances, mappings, constraints
- Dix '04: Human-Computer Interaction
  - Very good general textbook
- Shneiderman '09: Designing the UI
  - Technology, interviews
- Nielsen '93: Usability Engineering (prototyping)
  - How to bring usability to the business



HCI Literature Sources: Books

- Buxton '07: Sketching User Experiences
  - Focuses on the early stages of product design
- Winograd (ed.) '96: Bringing Design To Software
  - Collection of articles, case studies
- Preece '07: Interaction Design
  - Design-focused textbook
- Boggridge '08: Designing Interactions
  - Coffee-table style, short essays about seminal digital product design
- See http://hci.rwth-aachen.de/hcibooks for a more detailed list with descriptions