HCI Study Guide

What is HCI?

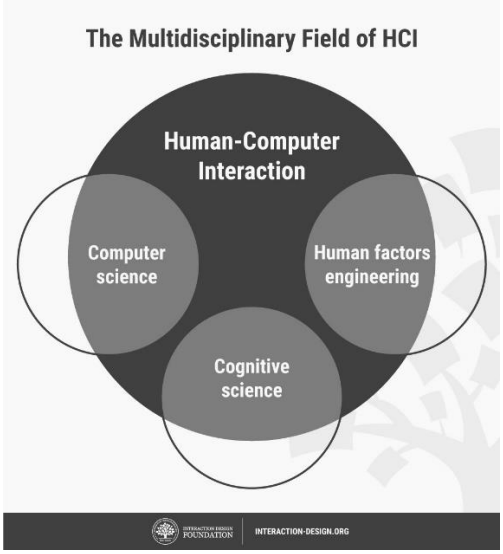
Definition

Human-computer interaction (HCI) is the creation and/or study of interactive systems that are (a) technically innovative, *and* (b) assessed primarily according to their value toward addressing user goals.

// According to Alex Quinn

"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

// Credit: Thomas T. Hewette et al. (1992) ACM SIGCHI Curricula for Human-Computer Interaction. https://dl.acm.org/citation.cfm?id=2594128



// Credit: © Glasbergen, All Rights Reserved https://www.interaction-design.org/literature/topics/human-computer-interaction

Types of contributions

- Development or Refinement of Interface Artifacts or Techniques
- Understanding Users
- Systems, Tools, Architectures and Infrastructure
- Methodology
- Theory
- Innovation, Creativity, and Vision
- Argument
- Validation and Refutation
- Public Policy

// Credit: https://chi2017.acm.org/contributions.html (authors unknown)

Usability metrics ★★★★★

L Learnability

E Efficiency

M Memorability

E Error resistance

S Satisfaction

// Credit: Nielsen, J. (1996). Usability metrics: Tracking interface improvements. IEEE Software, 13(6), 12-13. doi:http://dx.doi.org/10.1109/MS.1996.10031

Human capabilities

Fitts' Law

$MT = a + b\ ID$
$MT = a + b \log_2 (2A/W)$
ID is the index of difficulty for a target
A = amplitude (distance to move)
W = width, relative to the axis of motion
2 is to ensure that the ID is always positive

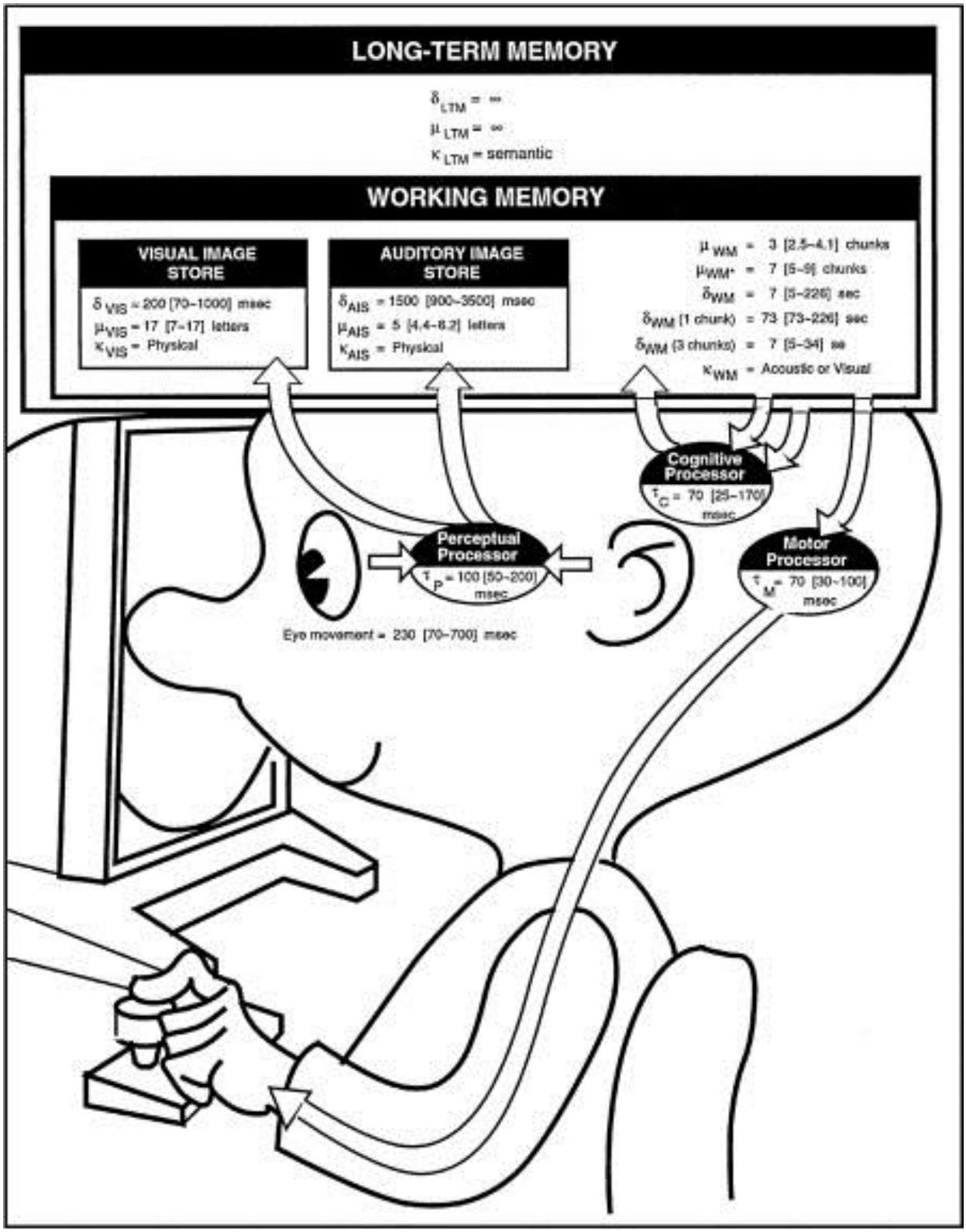
Miller's Law

Memory span is limited to 7 ± 2 chunks
“Memory span” is the number of items a person can remember for short periods of time and repeat back correctly
“Chunks” are the mental units we use to remember things.
Miller’s Law relates to short-term memory

Hick's Law

$RT=a+b \log_2 (n)$
RT is the the reaction time to make a choice
a, b depends on the type of task and is determined experimentally
n is the number of choices

Model Human Processor (MHP)



// Credit: Stuart K. Card, Thomas P. Moran, Alan Newell. The Model Human Processor: An Engineering Model of Human Performance. 1986.

How do we design?

User goals ★★★★★★★★★★★★

User goals are goals (outcomes) that the user wants to achieve—*independent of any solution*.

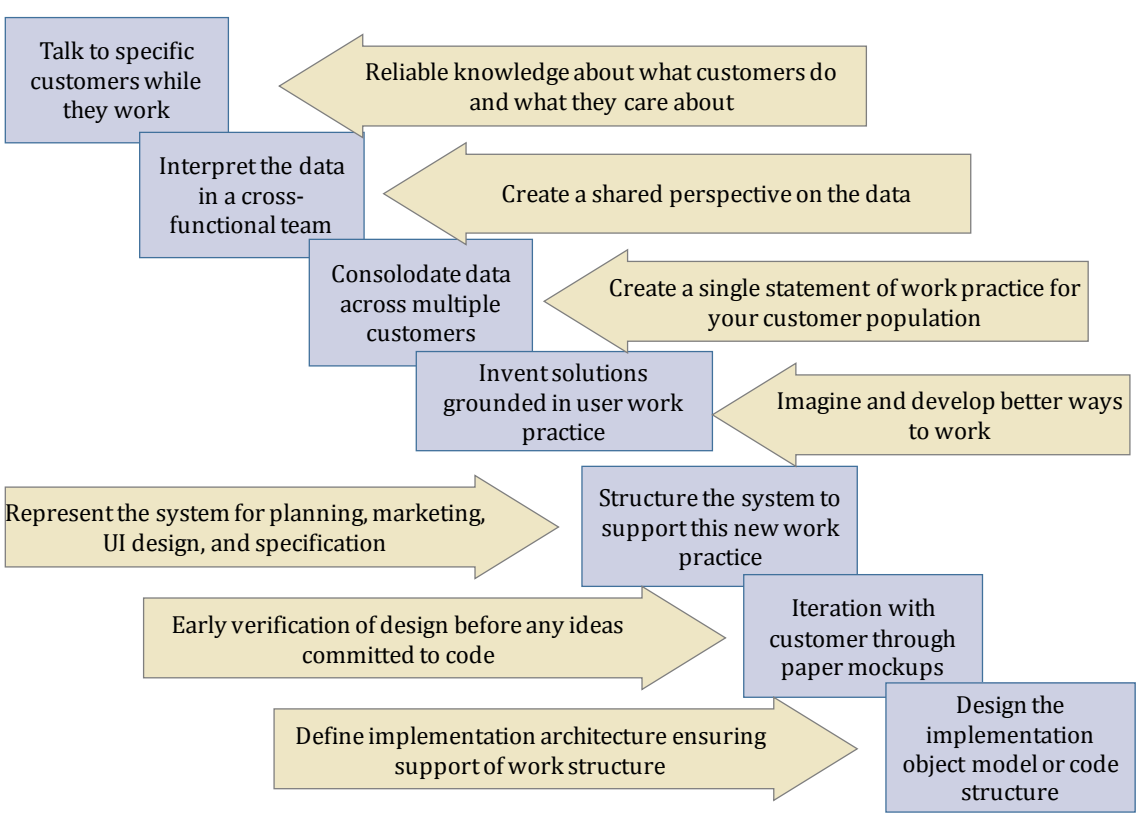
User goals should *not* include:

- Engineering goals (e.g., speed, thickness, and other technical specifications)
- Design goals (e.g., learnability, efficiency, memorability, error resistance, satisfaction, etc.)

Note: If you understand—and follow—the 2-3 sentences above, you know enough to ensure that your creations actually help users (rather than just adding more stuff to their lives). There isn't much more that *needs* to be said about it, but if you really want to read something, see *About Face 3.0: The Essentials of Interaction Design* (by Alan Cooper et al., ISBN 978-0-470-08411-3), chapter 1, pages 13-17. I learned the concept from an earlier edition of that book. It also distinguishes "user goals" from "activities" and "tasks" but that is mainly to ensure that user goals remain independent of any solution.

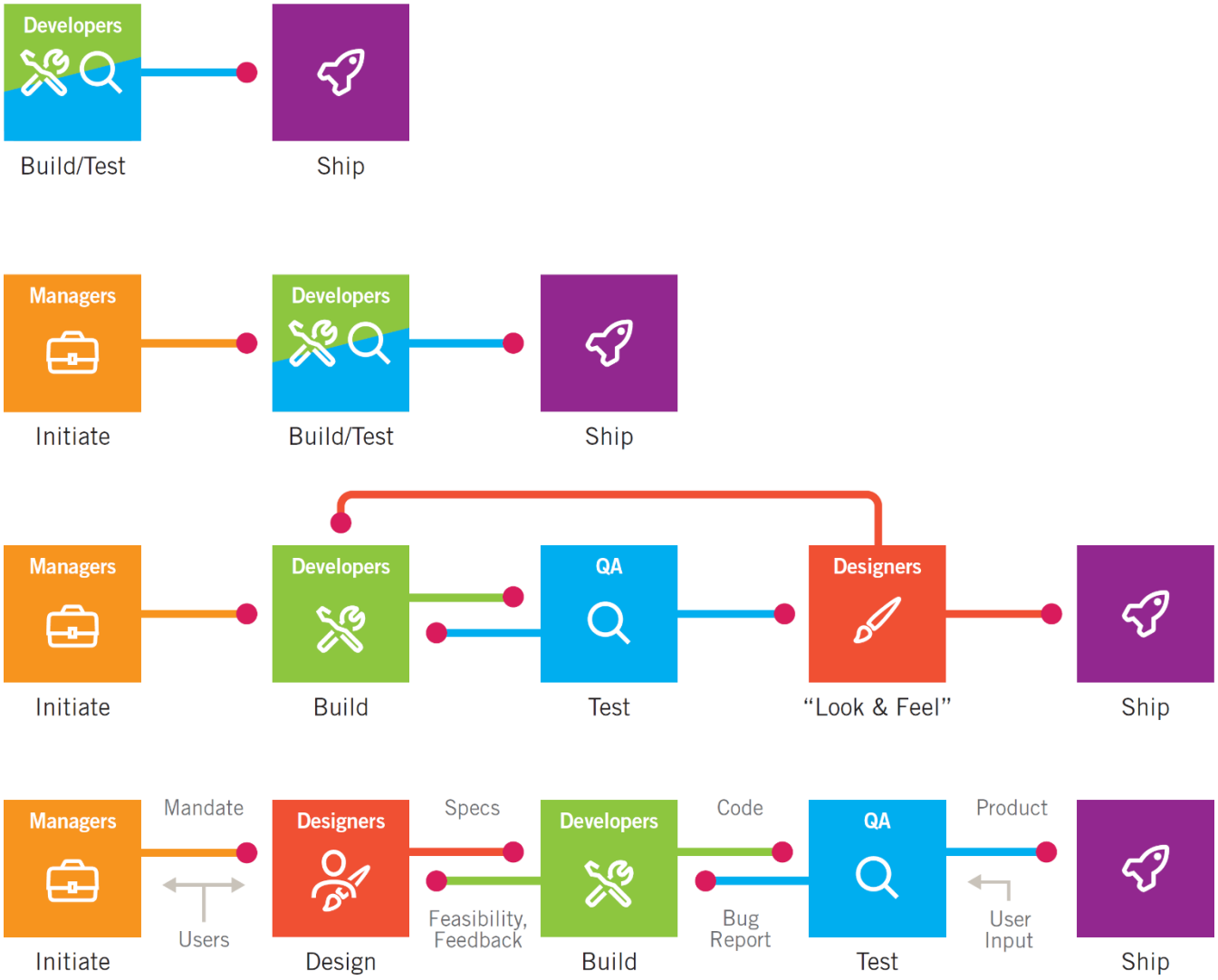
// Credit: © Alan Cooper et al. (2014) About Face: The Essentials of Interaction Design. ch. 1, pp.13-17
https://ebookcentral.proquest.com/lib/purdue/detail.action?docID=1762072

Contextual design



// Credit: © Hugh Beyer, Karen Holtzblatt. Contextual Design. Interactions Magazine. 1999. All Rights Reserved.
https://dl.acm.org/citation.cfm?id=291229 Image enhanced by Alexander J. Quinn.

Design to deployment



// Credit: © Alan Cooper et al. (2014) About Face: The Essentials of Interaction Design. ch. 1, p. 8, fig. 1-2
https://ebookcentral.proquest.com/lib/purdue/detail.action?docID=1762072

Concepts

See the reading (ch. 1) for explanations of terms.

- industrial design
- interaction design
- experience design
- human-centered design
- affordances
- signifiers
- constraints
- mappings
- feedback
- conceptual model

Constraints

See The Design of Everyday Things (chapter 4) for explanations of terms not defined here

- physical constraints
  - forcing functions - “actions are constrained so that failure at one stage prevents the next step from happening”
  - interlock - “forces operations to take place in proper sequence”
  - lock-in - “keeps an operation active, preventing someone from prematurely stopping it”
  - lock-out - “prevents someone from entering a space that is dangerous, or prevents an event from occurring”
- cultural constraints
  - conventions
- semantic constraints - “rely upon the meaning of the situation to control the set of possible actions”
- logical constraints

Errors

See The Design of Everyday Things (chapter 5) for more examples and discussion.

Types of errors

root cause analysis	“investigate the accident until the single, underlying cause is found”
five why's	Go ≥5 levels deep when asking why error happened
error	“all wrong actions”; in other words: “deviance from 'appropriate' behavior.”
deliberate violation	when “people knowingly take risks”
slip	“A slip occurs when a person intends to do one action and ends up doing something else.”
action-based slip	“wrong action is performed” Ex: “I poured some milk into my coffee and then put the coffee cup into the refrigerator. This is the correct action applied to the wrong object.”
memory-lapse slip	“memory fails, so the intended action is not done or its results not evaluated” Ex: “I forget to turn off the gas burner on my stove after cooking dinner.” Ex: “Making copies of a document, walking off with the copy, but leaving the original inside the machine”
mistake	“wrong goal is established or the wrong plan is formed”
rule-based mistake	“person has appropriately diagnosed the situation, but then decided upon an erroneous course of action: the wrong rule is being followed” "Rule-based mistakes occur in multiple ways: "• The situation is mistakenly interpreted, thereby invoking the wrong goal or plan, leading to following an inappropriate rule. "• The correct rule is invoked, but the rule itself is faulty, either because it was formulated improperly or because conditions are different than assumed by the rule or through incomplete knowledge used to determine the rule. All of these lead to knowledge-based mistakes. "• The correct rule is invoked, but the outcome is incorrectly evaluated. This error in evaluation, usually rule- or knowledge-based itself, can lead to further problems as the action cycle continues."
knowledge-based mistake	“problem is misdiagnosed because of erroneous or incomplete knowledge” Ex: “Weight of fuel was computed in pounds instead of kilograms” Note: “Knowledge-based behavior takes place when the situation is novel enough that there are no skills or rules to cover it.”
memory-lapse mistake	“there is forgetting at the stages of goals, plans, or evaluation” Ex: “A mechanic failed to complete troubleshooting because of distraction.” ⊂ action-based slips
capture slip	“instead of the desired activity, a more frequently or recently performed one gets done instead: it captures the activity” ⊂ action-based slips
description-similarity slip	“act upon an item similar to the target” Ex: If “both the laundry basket and the toilet bowl are containers, and if the description of the target was sufficiently ambiguous, such as 'a large enough container,' the [description-similarity] slip could be triggered.”
mode error	using controls that would be appropriate if the device were in a different state, but have the wrong meaning in the current state ⊂action-based slips
checklists	useful tool for preventing slips and memory lapses

Solutions to errors

- Analyze problems
  - root cause analysis - “investigate the accident until the single, underlying cause is found”
  - five why's – go ≥5 levels deep when asking why error happened
- Avoid
  - Add constraints. "Use the power of natural and artificial constraints: physical, logical, semantic, and cultural. Exploit the power of forcing functions and natural mappings.” *See the section on constraints above,*
  - Visibility (e.g., system status, needed knowledge, etc.). “ensure that all the relevant information is continuously available” -- memory-lapse mistakes
  - Start with a “good understanding of the situation” and develop an “appropriate conceptual model” to avoid knowledge-based mistakes
  - Checklists to avoid slips and memory-lapses)
  - Sensibility checks. “Does the action pass the 'common sense' test?”
- Mitigate errors when they occur
  - “Don’t treat the action as an error; rather, try to help the person complete the action properly. Think of the action as an approximation to what is desired.”
  - Undo. “Make it possible to reverse actions—to “undo” them—or make it harder to do what cannot be reversed.”
  - “Make it easier for people to discover the errors that do occur, and make them easier to correct.”
  - Undo
  - “Confirmation and error messages”
- General (higher level thinking)
  - Resilience engineering “with the goal of designing systems, procedures, management, and the training of people so they are able to respond to problems as they arise. It strives to ensure that the design of all these things—the equipment, procedures, and communication both among workers and also externally to management and the public—are continually being assessed, tested, and improved.”
  - "Put the knowledge required to operate the technology in the world. Don’t require that all the knowledge must be in the head. Allow for efficient operation when people have learned all the requirements, when they are experts who can perform without the knowledge in the world, but make it possible for non-experts to use the knowledge in the world. This will also help experts who need to perform a rare, infrequently performed operation or return to the technology after a prolonged absence.”
  - Swiss cheese model → (a) “Add more slices of cheese.” (b) “Reduce the number of holes (or make the existing holes smaller).” (c) “Alert the human operators when several holes have lined up.”
  - "Bridge the two gulfs, the Gulf of Execution and the Gulf of Evaluation. Make things visible, both for execution and evaluation. On the execution side, provide feedforward information: make the options readily available. On the evaluation side, provide feedback: make the results of each action apparent. Make it possible to determine the system’s status readily, easily, accurately, and in a form consistent with the person’s goals, plans, and expectations.”

Design reasoning

What is design (versus science)?

Scientific design - "modern, industrialized design-as distinct from pre-industrial, craft-oriented design-based on scientific knowledge but utilizing a mix of both intuitive and nonintuitive design methods."

Design science - "addresses the problem of determining and categorizing all regular phenomena of the systems to be designed, and of the design process"

Design (discipline) - "design studied on its own terms, and within its own rigorous culture. It can mean a science of design based on the reflective practice of design: design as a discipline, but not design as a science. This discipline seeks to develop domain-independent approaches to theory and research in design.32 The underlying axiom of this discipline is that there are forms of knowledge special to the awareness and ability of a de- signer, independent of the different professional domains of design practice."

// Credit: Nigel Cross. Designerly Ways of Knowing: Design Discipline Versus Design Science. Design Issues. 2001

Questions Options Criteria

Takeaway: When you're stuck in design discussions, make a multiple choice quiz that you complete together. With each question, your design space should get narrower.

- **design rationale** - "representation for explicitly documenting the reasoning and argumentation that make sense of a specific artifact."
- **design space analysis** - "creates an explicit representation of a structured space of design alternatives and the considerations for choosing among themdifferent choices in the design space resulting in different possible artifacts. Thus, a particular artifact is understood in terms of its relationship to plausible alternative artifacts."

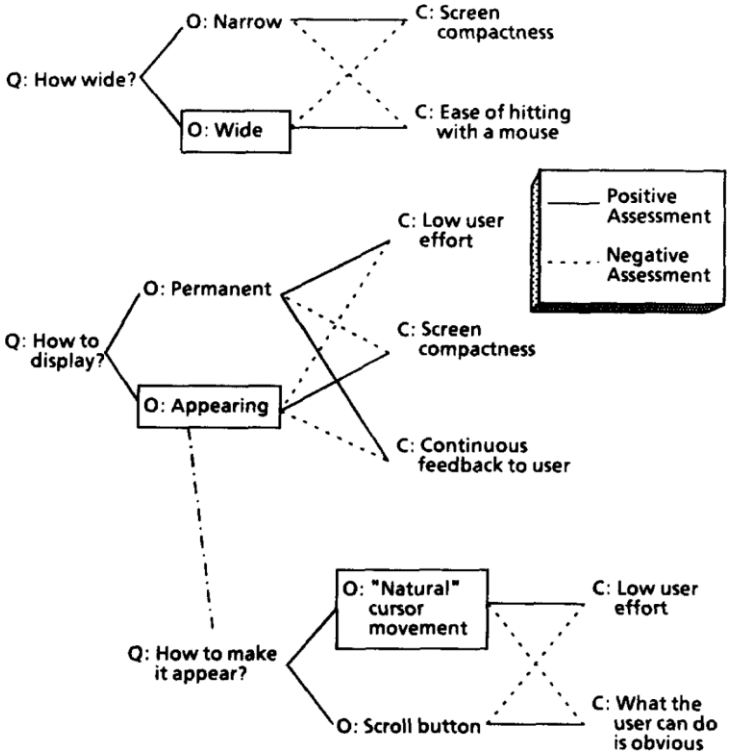


Figure is from MacLean et al. paper (referenced below), p. 209, fig. 3.

// Credit: MacLean et al. Questions Options and Criteria: Elements of Design Space Analysis. 1991.

Cognitive dimensions of notations

Examples and more detailed definitions are given in the reading.

1. "Viscosity: resistance to change."
2. "Visibility: ability to view components easily."
3. "Premature commitment: constraints on the order of doing things."
4. "Hidden dependencies: important links between entities are not visible."
5. "Role-expressiveness: the purpose of an entity is readily inferred. "
6. "Error-proneness: the notation invites mistakes and the system gives little protection."
7. "Abstraction: types and availability of abstraction mechanisms."
8. "Secondary notation: extra information in means other than formal syntax."
9. "Closeness of mapping: closeness of representation to domain."
10. "Consistency: similar semantics are expressed in similar syntactic forms."
11. "Diffuseness: verbosity of language."
12. "Hard mental operations: high demand on cognitive resources."
13. "Provisionality: degree of commitment to actions or marks."
14. "Progressive evaluation: work-to-date can be checked at any time."

Reminder: Cognitive dimensions of notations are not usability heuristics!

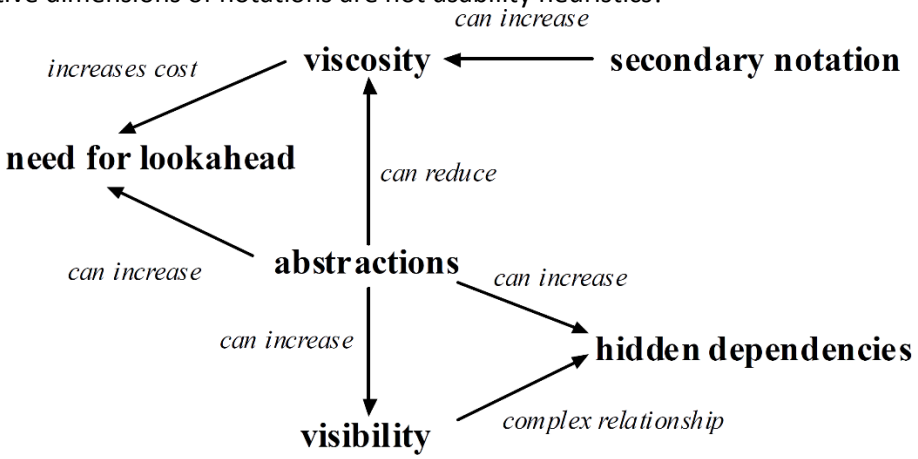


Figure is from Blackwell et al. paper (referenced below), p. 11, fig. 3.

// Credit: Alan Blackwell et al. Notational systems–the cognitive dimensions of notations framework. 2003. http://www.cl.cam.ac.uk/~afb21/publications/BlackwellGreen-CDsChapter.pdf[See diagram]

Experiment design

Prototyping

Mockups vs. Wireframes vs. Prototypes

- **Wireframes** are "low-fidelity" and show the paths through the system and the most important information presented or collected in each view.
- **Mockups** are "mid-fidelity" and should look like a screenshot.

- **Prototypes** may be "low or high fidelity" but should enable some limited interaction testing by team members and design informants (e.g., study participants, stakeholders participating in the design, etc.). Prototypes should be disposable—not just an early version of your eventual implementation.

► Reminder: Prototypes are not the same as alpha implementations. Prototypes should be built with the lowest effort means possible—i.e., crappy is good—with just enough fidelity and interactivity to test the idea.

Brainstorming

Before brainstorming	
Prepare the group	• "The . . . leader should be trained in advance for his function. Ideally he (the leader) should have taken a course in creative problem solving. At least, he should have assiduously studied problem-solving" (p. 172). • Define roles and responsibilities (i.e., leadership, group members). • Evaluate candidates for group membership based on the nature of the task, as well as self-starters, a mix of gender, styles, and content expertise. • Group size should be between 5–7 participants for each facilitator. • Invite participants and provide them with role definitions and key background information not less than 2 days before meeting. Include some examples of the type of ideas desired • Provide training in the guidelines before the meeting, including a thorough orientation of participants unfamiliar with brainstorming. • Provide for individual ideation before the session.
Prepare the task	• "The leader's first job is to process the problem . . . definition of aim is often half the battle" (pp. 172–173). • Provide a clearly focused problem statement (i.e., simple and specific, not complex or compound). • Select tasks that require many, varied, and unusual ideas. • Prepare a list of idea stimulating questions.
Prepare the environment	• "The spirit of a brainstorming session is important" (p. 157). • Ensure that, during the session, members have the same rank or level of power. • Display and prepare to reinforce the guidelines. • Prepare idea-spurring questions.
During brainstorming	
Dealing with judgment	• "Criticism is ruled out. Adverse judgment of ideas must be withheld until later" (p. 156). • "Freewheeling is welcomed. The wilder the idea the better; it is easier to tame down than to think up" (p. 156). • The session should start with an explanation of the task and a short Q&A. Explain guidelines in familiar language, using informal words. • Enforce guidelines gently, but firmly. •

Maintaining group commitment	• "Combination and improvements are sought . . . participants should suggest how ideas of others can be turned into better ideas" (p. 156). • Leaders should come prepared with ideas to submit during the dry periods. Extend effort by setting challenging quotas. • Provide incubation breaks. • The session lasts for 30–45 minutes. Reward all ideas with receptiveness. •
Enhancing the process structure	• "Quantity is wanted. The greater the number of ideas, the more the likeli- hood of useful ideas" (p. 156). • Collect individual idea lists before the session starts. • Each person makes notes of any ideas that they have until they can offer them. Have a recording secretary (or two if necessary). • Have a "warm-up" session, with exercises that practice and reinforce key skills and principles. • Tape record the session. •
After Brainstorming	
Follow-through	• "After brainstormers have slept on a problem, they sometimes generate the most valuable of all ideas" (p. 198). • Combination, elaboration, etc. should be used to further develop ideas. • Seek to reconsider the silliest ideas, and look for how they might be modified to create the best solution. • Ask for afterthoughts—additional individual ideation. • Provide feedback and thank you.
Evaluation	• "It is usually wise to have the final evaluation done by those directly responsi- ble for the problem" (p. 200). • Use a separate session dedicated to evaluating ideas ("Ideas should always be screened and otherwise processed—with a smaller and different group" [p. 200]). • Practical (e.g., pilot) testing is a most desirable method of verification.
Implementation	• "It is for want of imagination in their application, rather than in their means of acquisition, that they (creative ideas) fail. The creative process does not end with an idea—it only starts with an idea" (p. 197). • It takes 4 tons of ore to get 1 oz. of gold.

Table is reproduced from paper by Scott G. Isaksen et al. (see below), which quotes heavily from a paper by Alex F. Osborn.

// Credit: Alex F. Osborn. Applied imagination: Principles and procedures of creative problem solving (3rd Rev. ed.). 1963.

// Credit: Scott G. Isaksen et al. A Reexamination of Brainstorming Research: Implications for Research and Practice. 2005.

Usable security

- Path of least resistance. Match the most comfortable way to do tasks with the least granting of authority.
- Active authorization. Grant authority to others in accordance with user actions indicating consent.
- Revocability. Offer the user ways to reduce others' authority to access the user's resources.
- Visibility. Maintain accurate awareness of others' authority as relevant to user decisions
- Expected Ability. Maintain accurate awareness of the user's own authority to access resources.
- Trusted Path. Protect the user's channels to agents that manipulate authority on the user's behalf.

- Expressiveness. Enable the user to express safe security policies in terms that fit the user's task
- Relevant boundaries. Draw distinctions among objects and actions along boundaries relevant to the task
- Identifiability. Present objects and actions using distinguishable, truthful appearances
- Foresight. Indicate clearly the consequences of decisions that the user is expected to make.

Web development

We had light walk-throughs of HTML, JavaScript (ES6), React.js, Model-View Controller (MVC).

User studies

Evaluation methods

- Lab study — in-person
  - think aloud protocol
- Log analysis — remote
  - A-B testing
- Remote observation — remote
  - monitor screen
- Ethnography — in-person, in-situ
  - embed with users for extended period of time
- Interviews — in-person
  - e.g., contextual interview
- Cognitive walk-through — no users
- Heuristic evaluation — no users

Heuristic Evaluation

- Evaluators are experts
- ≥ 5 evaluators
- 1-2 hours to examine application
- Rate severity on a controlled scale

Severity scale

0. I don't agree that this is a usability problem at all
1. Cosmetic problem only: need not be fixed unless extra time is available on project
2. Minor usability problem: fixing this should be given low priority
3. Major usability problem: important to fix, so should be given high priority
4. Usability catastrophe: imperative to fix this before product can be released

// Credit: https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

Heuristics from Jakob Nielsen

- Match the real world
- Consistency & standards
- Help & documentation
- User control & freedom
- Visibility of system status
- Flexibility & efficiency

- Error prevention
- Recognition, not recall
- Error reporting, diagnosis, and recovery
- Aesthetic & minimalist design
- … and/or your own heuristics

// Credit: Jakob Nielsen
https://www.nngroup.com/articles/ten-usability-heuristics/

Ben Shneiderman's Eight Golden Rules

- Consistency
- Shortcuts
- Feedback
- Dialog closure
- Error prevention and easy recovery
- Undo
- User in control
- Reduce short-term memory load

// Credit: Ben Shneiderman.
https://www.cs.umd.edu/users/ben/goldenrules.html
Via http://web.mit.edu/6.813/www/sp18/classes/17-heuristic-evaluation/

Validity

External validity - Are results generalizable?

Ecological validity - Will results be applicable to real-life (i.e., non-research) settings?

Internal validity - Does test infer causality (to some degree)?

Construct validity - Does test measure what it claims to?

Content validity - Does test measure (sufficiently) all facets of the construct?

Face validity - Does test appear to measure what it is intended to test?

Criterion validity - Does outcome of test agree with other measures of the same construct?

Concurrent validity - Does test agree with other measures of the same construct taken at the same time?

Predictive validity - Does test predict other measures of the same construct in the future?

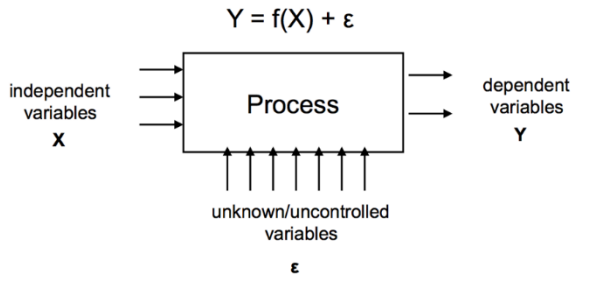
Discriminant validity - Does test differentiate things that are supposed to be different?

Example: The predictive validity of a cognitive test for job performance is the correlation between test scores and, for example, supervisor performance ratings.

Credit: Some of this is adapted from Wikipedia.org/wiki/Validity_(statistics)

Experimental methods

Experiment



Hypothesis

Hypothesis should be...

- meaningful
- testable
- falsifiable
- precise

Validity

- **External validity** - Are results generalizable?
  - **Ecological validity** - Will results be applicable to real-life (i.e., non-research) settings?
- **Internal validity** - Does test infer causality (to some degree)?
- **Construct validity** - Does test measure what it claims to?
  - **Content validity** - Does test measure (sufficiently) all facets of the construct?
  - **Face validity** - Does test appear to measure what it is intended to test?
  - **Criterion validity** - Does outcome of test agree with other measures of the same construct?
    - **Concurrent validity** - Does test agree with other measures of the same construct taken at the same time?
    - **Predictive validity** - Does test predict other measures of the same construct in the future?
- **Discriminant validity** - Does test differentiate things that are supposed to be different?

Designs

- Between subjects
  - Need more subjects
  - Eliminates variation due to ordering, learning
  - Choose if learning/ordering effects are a threat
    - Or if task is very long
- Within-subjects
  - Eliminates variation due to user differences
  - User sees all conditions
  - Choose if differences among people are a threat
    - Or if task is very short

Counterbalancing

- Randomize order of treatments that would affect results
- Latin Squares
  - Each condition appears in a given position a fixed number of times

G1	G2
A	B
B	A

2x2

G1	G2	G3
A	C	B
B	A	C
C	B	A

3x3

G1	G2	G3	G4
A	D	C	B
B	A	D	C
C	B	A	D
D	C	B	A

4x4

... etc.

Types of variables

	Discrete	Ordered	Proportional	Zero	# of values
Continuous	No	Yes	No	*	*
Categorical	Yes	No	No	No	*
Ordinal	Yes	Yes	No	No	*
Interval	Yes	Yes	Yes	No	*
Ratio	Yes	Yes	Yes	Yes	*
Dichotomous	Yes	*	n/a (?)	*	2
Boolean	Yes	Yes	Yes	Yes	2 (T/F)

Survey methods

See slides for more.

- Appropriate use
- Constructs
- Pre-validation
- Sampling
- Recruiting
- Questionnaire design
- Types of questions
- Types of variables
- Close-ended vs. open-ended
- Biases
- Other hazards
- Established questinnaires
- UI design considerations
- Commercial services
- Launch tips

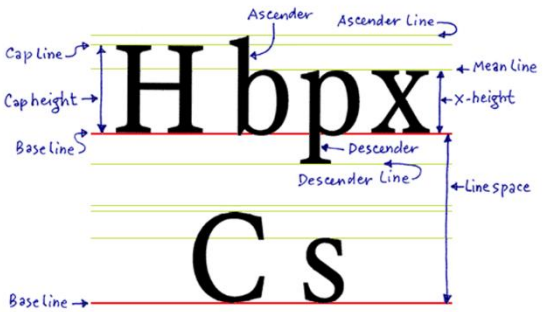
Visual design

Type

Typography in CSS

Relative to text	Relative to viewport	Absolute
em	vh	px
ex	vw	mm
ch	vmin	cm
rem	vmax	in
		pt
		pc
		mozmm

Font metrics



Credit: Ivo Vynckier, http://www.how-ocr-works.com/images/resolution.html

Typography in CSS

font-family	letter-spacing	text-transform
font-size	word-spacing	text-shadow
font-weight	line-height	vertical-align
font-style	text-align	white-space
font-variant	text-decoration	direction
font	text-indent	unicode-bidi

Gestalt Principles

Gestalt: Invariance

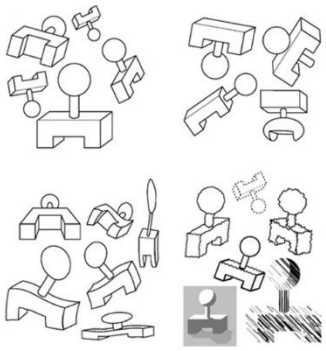


image: public domain; See https://en.wikipedia.org/wiki/Gestalt_psychology

Gestalt: Multistability

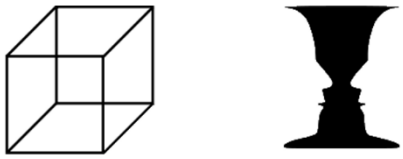


image: public domain; See https://en.wikipedia.org/wiki/Gestalt_psychology

Gestalt: Reification

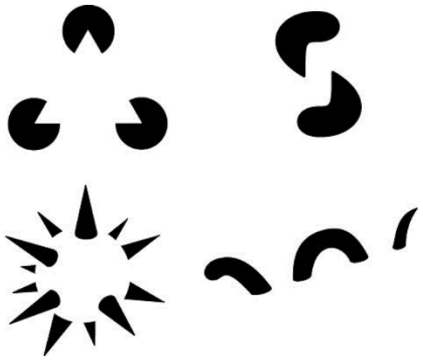


image: public domain; See https://en.wikipedia.org/wiki/Principles_of_grouping

Gestalt: Proximity

image: public domain; See https://en.wikipedia.org/wiki/Principles_of_grouping

Gestalt: Similarity

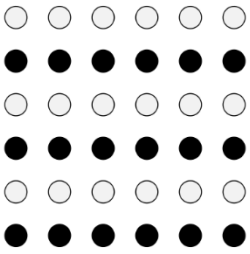


image: public domain; See https://en.wikipedia.org/wiki/Principles_of_grouping

Gestalt: Closure