Wisdom is not the product of schooling but the lifelong attempt to acquire it.
- Albert Einstein

Unusable at any Bandwidth
The Main Message of Chapter 2

software with poorly designed user interfaces sometimes has the same deadly or poisonous effects as car design failures or pollutants
Claims by Lee

- Software errors do not have to be mistakes.

- Even when computer software correctly does what it is supported to do, it can cause problems.

- It can encounter situations unanticipated by whoever programmed it.

- **Software will only do whatever it is told to do. (???)**
### Design Disasters versus Implementation Disasters

<table>
<thead>
<tr>
<th>World</th>
<th>Model</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Diagram" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Upstream**
- Design

**Downstream**
- Engineering

- **upstream: world → model / specification**
  - ill-defined problem
  - integration of problem framing and problem solving
  - collaboration and communication between different stakeholders
  - failure leads to *design disasters* (wrong problem is solved)

- **downstream: model / specification → implementation / system**
  - well-defined problem
  - dealing with difficult technical problems
  - creating reliable code
  - failure leads to *implementation disasters* (wrong solution to the right problem)
Example from Lee: Unfriendly Skies

Paradoxes of Increased Reliance on Software-Controlled Computers in Airplanes

- as flight systems become more reliable, the anticipation of reliability may make pilots less able to spot errors

- pilots do what the software directs them to do, even if it’s wrong

**related claim:** anticipation of trust $\rightarrow$ learners will believe information from Wikipedia (from teachers, ....)

**challenges:**
- AI (Artificial Intelligence) $\rightarrow$ IA (Intelligence Augmentation)
- computers should assist human decision making not replace it
Different Relationships between Automation and Human Involvement

High Automation

Autonomous Operation
Management by Exception
Management by Consent
Management by Delegation
Shared Control
Assisted Manual Control
Direct Manual Control

Low Automation

High Human Involvement
Low Human Involvement
Example from Lee: 1988 Shooting Down of an Iranian Airbus by the USS VINCENNES

- **American Psychological Association (APA) Congressional Testimony:**
  “predictable failings in human judgment under stressful conditions, compounded by complex technology, clearly contributed to the accident”

- **Contributing factors:**
  - Inaccurate perceptions about the airline’s flight path (most striking error: judgment that the flight was descending for several minutes in which it never failed to ascend)
  - Communications problems in the Combat Information Center (CIC; AEGIS)
  - Design limitations in panel displays
  - Overreliance on computer systems
  - Faulty design making under stressful circumstance (e.g.: under time stress, people may treat other people’s judgment as fact → scenario fulfillment)

- **Recommendation:** Research support on human performance and decision making under stress
Reliance on the “Right” Kind of Technology
Petroski, H. (1985) To Engineer Is Human: The Role of Failure in Successful Design
From Information Famine to Information Glut

“If computers are to be helpful to us at all, it must not be in producing more information—we already have enough to occupy us from dawn to dusk—but to help us to attend to the information that is the most useful or interesting or, by whatever criteria you use, the most valuable information.” — Herbert Simon

- Information Glut
  - over 100 million books in the Library of Congress
  - over 500 daily newspapers in the USA
  - number of professional journals is growing at a rapid rate
  - McGuckin Hardware Store
  - High-functionality applications
  - Reuse libraries
  - World Wide Web
Human-Centered: More Than User Interfaces

- **human-computer interaction is more than user interfaces**
  Applying the Macintosh style to poorly designed applications and machines is like trying to put Béarnaise sauce on a hot-dog! (A. Kay)

- **make systems useful and usable**
  If ease of use was the only valid criterion, people would stick to tricycles and never try bicycles. (D. Engelbart)

- **support human problem-domain interaction**
  Interfaces get into the way. I don't want to focus my energies on an interface. I want to focus on the job. (D. Norman)
“Human-Centered” — Some Objectives

- the system solves a **real human need**

- the system is well integrated into **real practice** (i.e.: the human spends too much time adapting to the system, rather than the system fitting into the realities of the context in which it exists)

- the user is able to **focus on the task** (i.e., the interface is the task; support for human problem-domain interaction)

- system should be **fun to use**
“Human-Centered” — Some Objectives (continued)

- achieve **external simplicity** with internal complexity → human-centered automation is not a call for less technology

- people need **contextualized information**

- **exceptions** are normal in work processes

- **co-evolution / co-adaptation**: “people not only adapt to their systems, they adapt their systems to their needs”

- break down the walls that **separate designers and users**
Human-Centered: utility = \textit{value} / \textit{effort}

- approaches/ techniques to \textit{increase value}
  - 
  - 

- approaches/ techniques to \textit{decrease effort}
  - 
  - 
Widely Help Assumptions and Beliefs

- reality is “user-friendly”

- the user interface is the major problem for HCI research

- most users are interested in computers per se, rather than in their tasks

- “high-tech scribes” and “complete idiots” are the primary computer users

- “experts” exist (complex systems can and will be completely learned)

- information is the scarce resource
Useful versus Usable → Useful and Usable

- **usable (as main objective):**
  - novices
  - limited functionality
  - low threshold to get started
  - walk-up and use
  - “experts” exist
  - understandable model of the complete system can be developed
  - examples: ATMs, VCRs, MacIntosh,

- **useful (as main objective):**
  - skilled users
  - broad functionality
  - high ceiling for skilled users
  - no “experts” (learning on demand is a necessity rather than a luxury)
  - no complete models
  - end-user modifiability, programmability
  - examples: MS-Word, Excel, Mathematica, Photoshop
Beyond User-Friendly: Low Threshold and High Ceiling
Useful Systems are High Functionality Applications (HFCS)

Problems with HFCS

- users do not have **well-formed goals and plans**
- users do not know about the **existence** of components
- users do not know how to **access** components
- users do not know **when** to use component
- users do not understand the **results** that components produce for them
- users cannot combine, adapt, and modify components according to their **specific needs**
Commercial Trends

- more **functionality**

- more **scaffolding** → Example: Auto-Correct in MS-Word

- incremental **learnability**

- increase in **end-user control**:
  - embedded programming languages, programmable applications
  - more flexibility and possibilities for power users
The Skeptic’s Corner

- engineers who refer to usability issues as “soft” or to users as stupid
- system designers who believe that soon we will have an automatic interface design program
- usability “specialists” who superficially edit product releases as they are about to be shipped
- designers who have never met or talked to someone who actually uses their products
- marketers who argue that most users don’t know what’s good for them in the first place