

CS-639 — Interaction Design Studio

**AI as Design Material — Designing
along the **Agency Spectrum****

Professor Bilge Mutlu

Today

- **A1 submitted** — congratulations. You've demonstrated fluency in design fundamentals + intelligence principles.
- **Phase 3 begins** — four weeks designing **with** intelligence properties
- **This week: Agency** — how much should the system do?

You've learned what intelligence enables. Now the question is: how much of it should you use?

The Central Design Question

When you add intelligence to an interface, you face a question that traditional design never asks:
How much should the system do on its own?

- Too little → intelligence adds no value
- Too much → users lose control, trust, and understanding
- The right answer → depends on the task, the user, and the stakes

Part 1: What Are the Options?

**A vocabulary for describing agency
— levels, stages, and thresholds**

Parasuraman et al. (2000)

A foundational framework for thinking about automation — not as a switch, but as a spectrum.¹

"Automation does not merely supplant but changes human activity and can impose new coordination demands on the human operator."

- **10 levels** of automation — from fully manual to fully autonomous
- **4 stages** of information processing — each can be automated independently
- Automation is not binary. It's a spectrum with meaningful thresholds.

¹ Parasuraman, Sheridan & Wickens (2000), "A Model for Types and Levels of Human Interaction with Automation," IEEE Trans. SMC

10 Levels of Automation

Parasuraman, Sheridan & Wickens (2000)



Two Critical Thresholds

Level 5 → 6: The Initiative Shift

Below 6, the human must act for anything to happen. At 6, the system acts **unless** the human intervenes.

Initiative transfers from human to system.

Autocorrect changes your text. You have a brief window to undo.

Level 7 → 8: The Transparency Shift

At 7, the system always tells you what it did. At 8, it tells you only if you ask.

The system starts making invisible decisions.

Gmail's spam filter tells you "23 messages filtered." Background app updates don't notify unless you check.

Products You Know — Mapped to Levels

Level	Label	Product	What Happens
1	Human does everything	Notepad	No suggestions, no assistance
2–3	Offers alternatives	Google Search	Offers ranked results; user chooses
3	Narrows to a few	Spotify Discover Weekly	Narrows millions of songs to 30
4	Suggests one option	Google Maps	Suggests best route; user can pick another
5	Executes if approved	Gmail Smart Reply	Drafts response; sends only if you tap
6	Executes; user can veto	Autocorrect	Changes text; brief window to undo
7	Executes; always informs	Gmail spam filter	Filters automatically; tells you after
8	Executes; informs if asked	Background app updates	Updates apps; tells you only if you check
9	Executes; informs if it decides	Nest thermostat	Adjusts temperature; may not explain why
10	Fully autonomous	ABS braking	Takes full control; no override possible

The Initiative Shift in Action

Autocorrect — Level 6

The system changes your text automatically. You see the change briefly and can undo it, but the window is short.

Design challenge: How do you give users enough time to notice and reject — without slowing them down?

Level 6 is where most design tension lives. The system is fast, but fast enough for the user to verify?



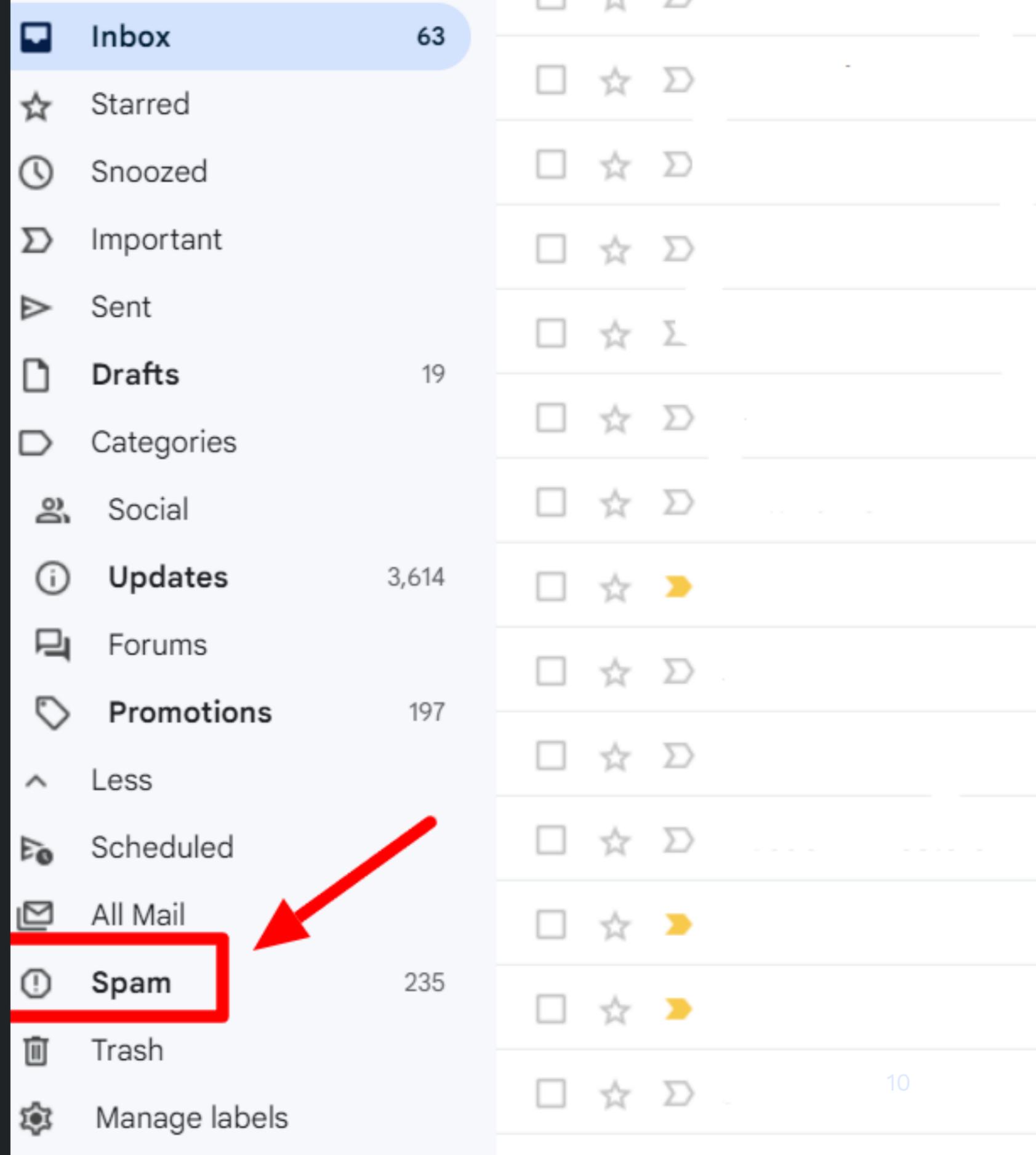
The Transparency Shift in Action

Gmail Spam Filter — Level 7

The system moves messages out of your inbox automatically and tells you: "23 messages in Spam."

Design challenge: How do you maintain trust when the user can't see every decision? What if the system is wrong?

At Level 7+, the user can't verify every decision. Trust depends on the system being right — and being transparent when it's wrong.



Four Stages of Information Processing

Parasuraman's second insight: every task has four stages. You can automate each independently.

Stage	What It Does	Design Question
Acquisition	Sensing, collecting, filtering data	What does the system gather?
Analysis	Pattern recognition, prediction	What does the system figure out?
Decision	Selecting an action	Who decides what to do?
Action	Executing the choice	Who carries it out?

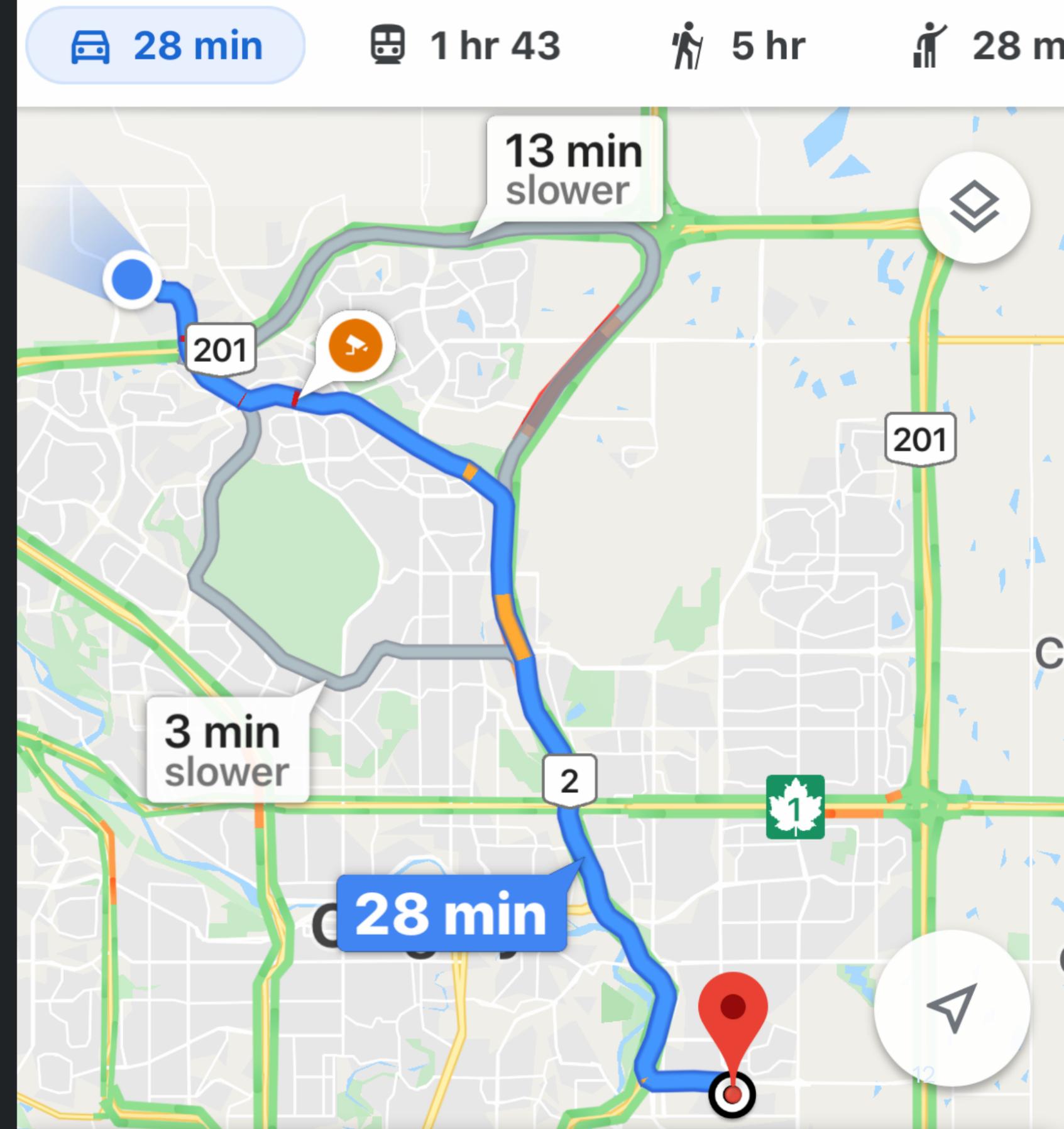
You can set **different automation levels** for each stage. This is not one slider — it's four.

One Product, Four Levels

Google Maps:

Stage	Level	What Maps Does
Acquisition	High (8–9)	GPS, traffic, satellite
Analysis	High (8–9)	Route calc, ETA, incidents
Decision	Medium (4)	Suggests route; user picks
Action	Low (1–2)	User drives

What would change if Maps were Level 8 on Decision? Level 10 on Action? That's autonomous driving.



Part 2: What Are the Risks?

Higher agency creates new failure modes — complacency, skill loss, and invisible errors

The Danger Zone: Levels 6–8

The most dangerous levels aren't 9–10, where humans are explicitly removed. They're **6–8**, where humans **think** they're monitoring but have actually disengaged.

Parasuraman calls this the "**out-of-the-loop**" problem.

- Attention drifts — the system handles it
- Skills degrade — you stop practicing
- When automation fails, you can't recover
- Higher automation → worse handoff

Design implication: Design for the **failure case**, not just the success case.



68
mph



Autosteer Unavailable for the Rest of This D
Hold Steering Wheel to Drive Manually

Adaptive Automation

The best intelligent systems don't stay at one level — they shift dynamically.

- **Grammarly:** Highlights errors (Level 3–4) → auto-applies simple fixes (Level 6)
- **Navigation apps:** Gentle suggestions on highways → firm commands in complex interchanges
- **Smart home:** Setup asks preferences (Level 4) → learned mode acts on its own (Level 8–9)

This connects to two material properties: **adaptivity** (system changes over time) and **delegation** (user states goals, system handles steps).

45 Clarity ⓘ Back to all suggestions X

CONCISENESS

tons of

It appears that **tons of** may be unnecessary in this sentence. Consider removing it.

Learn more

From the main... · Rephrase sentence

that's · Remove the phrase

if you want · Remove the phrase

Keep your topi... · Rephrase sentence

Tip: To "borro... · Rephrase sentence

This · Rewrite the sentence

Within second... · Rephrase sentence

is sorted · Rewrite the sentence

Each idea is so... · Rephrase sentence

more · Remove the phrase

This · Rewrite the sentence

84 Overall score See performance

Goals Adjust goals

GrammarlyGO AI text generation

All suggestions

Correctness 17 alerts

Clarity A bit unclear

Engagement Engaging

Delivery Slightly off

Style guide All good

15 Get Expert Writing He

Plagiarism

Part 3: How Do You Decide?

Four factors for choosing where a feature should sit on the spectrum

Choosing the Right Level

Parasuraman gives you the vocabulary. Now you need a way to **decide**.

Four factors determine where a feature should sit on the spectrum:

Factor	Lower agency (1–5)	Higher agency (6–10)
Error cost	Errors are costly or harmful	Errors are minor or recoverable
Reversibility	Actions are hard to undo	Actions are easily undone
Frequency	Rare, high-stakes decisions	Frequent, routine tasks
System confidence	System is often wrong	System is reliably accurate

Agency and Material Properties

Adaptivity

How much does the system learn and change?

Maps to **information stages** (acquisition + analysis).

How aggressively should the system update its behavior based on what it learns?

Delegation

How much does the system decide and act?

Maps to **action stages** (decision + action).

How much decision-making should the user hand over?

Part 4: How Do You Design for Your Choice?

Mental models, trust calibration, and the HAX design checklist

Mental Models and Trust at Higher Agency

As automation level increases, two design challenges intensify:²

Mental Models — At Level 7+, the system makes invisible decisions. Users need an accurate model of what it does and why.

- **Teach effective use** — in-context guidance
- **Let users teach the system** — preferences, corrections

Trust Calibration — At Level 8, a bad decision may go unnoticed. Higher agency demands stronger trust calibration.

- **Be upfront about limitations**
- **Use friction to avoid overreliance**
- **Provide rationales** — show where outputs come from

² Weisz et al. (2024), CHI '24

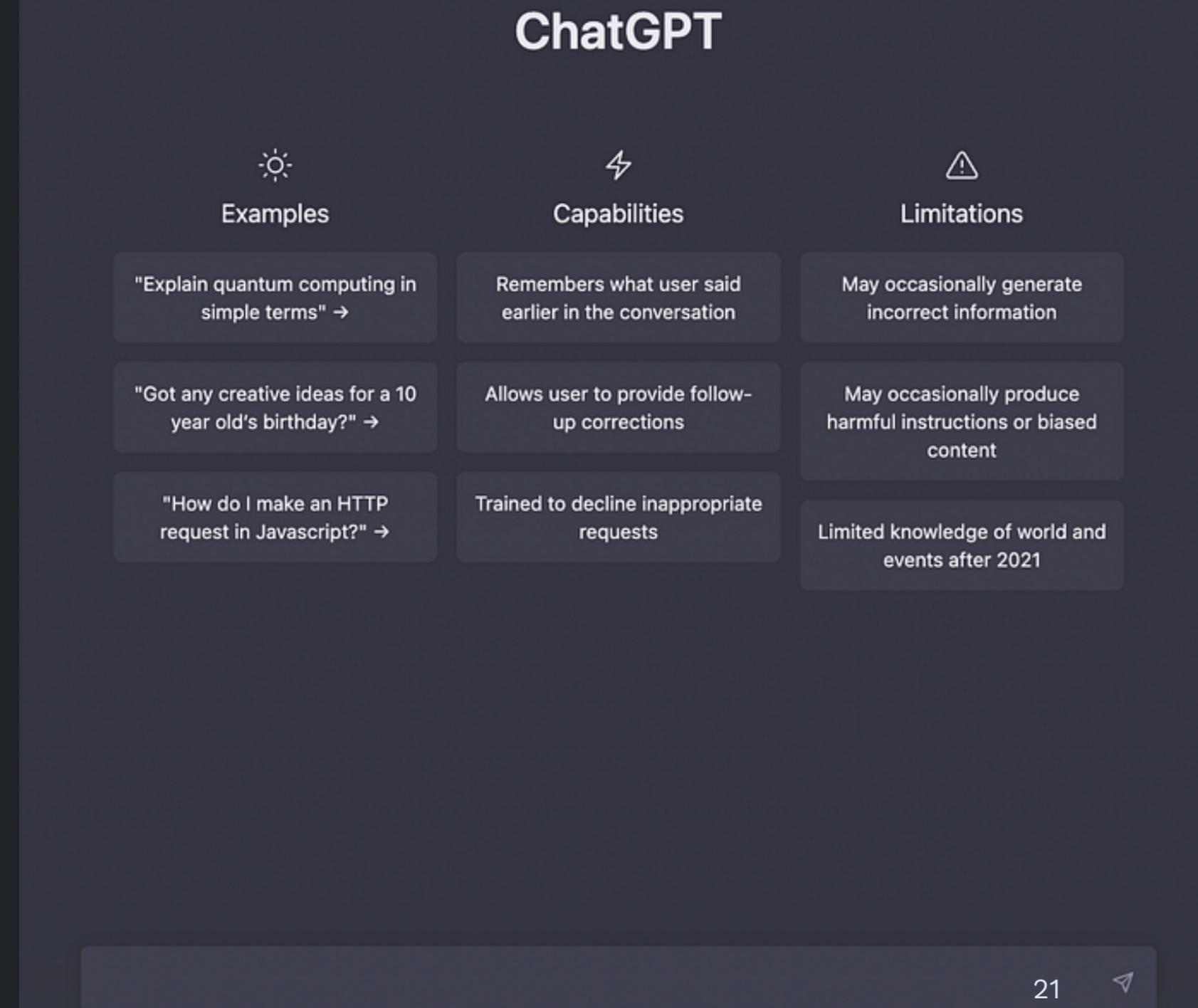
Case Study: ChatGPT's Original Onboarding

The original 3-column welcome screen (2022–2024):

- **Examples** — what to do
- **Capabilities** — how it works
- **Limitations** — when not to trust it

Three functions in one screen. 100M+ users saw it in two months.

Later removed. What does that tell you about mental model design?



Designing for Your Level: The HAX Toolkit

Once you've chosen an agency level, **how do you design well at that level?** Amershi et al. (2019) provide 18 guidelines, organized by when they matter most:³

Phase	Guidelines	Critical at...
Initially	G1–G2: Set expectations	All levels — but especially 5+
During Interaction	G3–G6: Be contextual, relevant	Levels 3–6
When Wrong	G7–G11: Support correction, explanation	Levels 5–8 (where users still monitor)
Over Time	G12–G18: Learn, adapt, provide controls	Levels 7+ (where system acts independently)

Higher agency → later-phase guidelines become **essential**, not optional.

³ [Microsoft HAX Toolkit](#) · [Amershi et al. \(2019\)](#)

1
INITIALLY

Make clear what the system can do

Help the users understand what the AI system is capable of doing.

2
INITIALLY

Make clear how well the system can do what it can do.

Help the user understand how often the AI system may make mistakes.

3
DURING INTERACTION

Time services based on context.

Time when to act or interrupt based on the user's current task and environment.

4
DURING INTERACTION

Show contextually relevant information.

Display information relevant to the users' current task and environment.

5
DURING INTERACTION

Match relevant social norms.

Ensure the experience is delivered in a way that users would expect, given their social and cultural context.

6
DURING INTERACTION

Mitigate social biases.

Ensure the AI system's language and behaviors do not reinforce undesirable and unfair stereotypes and biases.

 **INITIALLY**

 **DURING INTERACTION**

7
WHEN WRONG

Support efficient invocation.

Make it easy to invoke or request the AI system's services when needed.

8
WHEN WRONG

Support efficient dismissal.

Make it easy to dismiss or ignore undesired system services.

9
WHEN WRONG

Support efficient correction.

Make it easy to edit, refine, or recover when the AI system is wrong.

10
WHEN WRONG

Scope services when in doubt.

Engage in disambiguation or gracefully degrade the AI system's services when uncertain about a user's goals.

11
WHEN WRONG

Make clear why the system did what it did.

Enable the user to access an explanation of why the AI system behaved as it did.

 **WHEN WRONG**

12
OVER TIME

Remember recent interactions.

Maintain short-term memory and allow the user to make efficient references to that memory.

13
OVER TIME

Learn from user behavior.

Personalize the user's experience by learning from their actions over time.

14
OVER TIME

Update and adapt cautiously.

Limit disruptive changes when updating and adapting the AI system's behaviors.

15
OVER TIME

Encourage granular feedback.

Enable the user to provide feedback indicating their preferences during regular interaction with the AI system.

16
OVER TIME

Convey the consequences of user actions.

Immediately update or convey how user actions will impact future behaviors of the AI system.

17
OVER TIME

Provide global controls.

Allow the user to globally customize what the AI system monitors and how it behaves.

18
OVER TIME

Notify users about changes.

Inform the user when the AI system adds or updates its capabilities.

 **OVER TIME**

This Week

Reflection, studio challenge, and preparation for Wednesday

Reflection: From Theory to Design

Due before Wednesday | Graded: ✓ / ✓- / ✓+

This week's reflection is more academic. You'll work with **three key passages** from Parasuraman et al. For each one:

1. **Explain** the concept in your own words (2–3 sentences)
2. **Design** a concrete response — sketch or describe a feature at two automation levels, map a task across four stages, or design a safeguard

Choose one product domain and use it for all three passages.

This builds the bridge from theory to practice — the same bridge you'll cross in Wednesday's studio.

Wednesday Studio: Designing Across the Agency Spectrum

Design the **same task at three agency levels** — low, medium, and high — then use the four decision factors and HAX guidelines to find the right level.

- **Pick a domain and task** — your A1 app works well
- **Sketch three versions:** Low (Levels 1–3), Medium (Levels 4–6), High (Levels 7–9)
- **Analyze with four factors:** Error cost, reversibility, frequency, system confidence
- **Evaluate with HAX:** Which guidelines become harder to satisfy at higher agency?
- **Recommend:** What's the right level — justified by the four factors?

The test: at high agency, what happens when the system is wrong? If you haven't designed for that, your agency level is too high.

Before Wednesday

- **Explore:** Microsoft HAX Toolkit — browse the 18 guidelines and design patterns
- **Submit:** Reflection on Canvas (three passages + design responses)
- **Choose:** Your domain and task for Wednesday's studio
- **Think about:** Where does your task sit on the agency spectrum right now?

Come ready to design the same task at three different agency levels.

References

Recommended Reading:

- Parasuraman et al. (2000). "Types and Levels of Human Interaction with Automation" — IEEE Trans. SMC

Design Frameworks:

- Amershi et al. (2019). "Guidelines for Human-AI Interaction" — CHI '19
- Weisz et al. (2024). "Design Principles for Generative AI Applications" — CHI '24

Tools:

- Microsoft HAX Toolkit

Background:

- Horvitz (1999). "Principles of Mixed-Initiative User Interfaces" — CHI '99
- Holmquist (2017). "Intelligence on Tap"

Media Sources

Autocorrect | iOS Autocorrect Predictions | Gmail Spam Filter | Spotify Discover Weekly | Google Maps | Nest Thermostat | Tesla Autopilot | Grammarly | ChatGPT | Microsoft HAX Toolkit