THE COMPLETE GUIDE TO AGENTIC UX DESIGN

How to design for AI that acts and decides without you

-Shreyas Kalyanpad

What is Agentic UX?

Old apps waited for your click. AI agents act on their own.

UX (User Experience) means how easy an app is to use. **Agentic** means the AI can take actions by itself.

Example: Instead of searching for flights yourself, an AI agent finds, compares, and books them while you sleep. The designer's job has shifted from building buttons to designing trust.

The 4 Levels of AI Agency

Level	Description	Example
Level 1	Reactive: AI answers when you ask.	A basic chatbot.
Level 2	Assistive: AI suggests, but waits for your approval.	Gmail Smart Compose.
Level 3	Semi-Autonomous: AI handles multi-step tasks with occasional check-ins.	AI coding agents (like Cursor or GitHub Copilot Chat running tasks)
Level 4	Fully Agentic: AI plans and executes complex tasks end-to-end with no supervision.	AI that runs a full marketing campaign (research → content → posting → optimization)

Why Old Design Rules Break Down

Classic design rules were written for apps where *YOU* were in control. With agentic AI, the app acts first and asks later.

Three problems:

1. **The Visibility Problem:** In an AI running 50 background tasks, what does 'show what's happening' even mean?
2. **The Control Paradox:** The approve button nobody reads.
3. **The Accountability Gap:** When AI made 10 decisions that caused an error, whose responsibility is it?

The Review Paradox

"When verifying what the AI did is harder than doing it yourself, your 'human in the loop' is just a rubber stamp."

Designing genuinely easy audits, not just technically possible ones, is the real design challenge.

8 Principles of Agentic UX

1. **Start slow, earn trust:** don't hand over full control immediately
2. **Show reasoning, don't hide it:** make AI logic accessible on demand
3. **Signal confidence:** show how sure the AI is before it acts
4. **Always allow pause and undo:** every agentic action needs a stop button
5. **Build consent architecture:** users must be able to see and revoke AI permissions
6. **Design the audit interface:** a readable history of everything the AI did
7. **Design for failure:** agents will get it wrong; make recovery easy
8. **Preserve context:** users should return to a clear picture of what happened while they were away

6 Design Patterns That Work

- **Progress Ledger:** plain-language timeline of what the AI is doing right now
- **Sandbox Preview:** show users what *will* happen before the AI does it
- **Intent Confirmation:** Before a long task, confirm that the AI understood your goal correctly
- **Delegation Dial:** a simple control to set how much autonomy the AI has
- **Handoff Moments:** clear transitions when control passes to/from the AI

- **Consequence Labels:** mark every action as reversible, costly to reverse, or irreversible

Quick Audit Checklist

- [ ] Can users see what the AI is doing right now?
- [ ] Is the AI's reasoning accessible on demand?
- [ ] Can users pause or undo any AI action?
- [ ] Are irreversible actions clearly labelled?
- [ ] Is there a readable history of AI actions?
- [ ] Do users understand what the AI is allowed to do?
- [ ] Does the AI communicate its uncertainty?

The Opportunity

The teams investing in agentic UX today are building the trust advantage of the next 5 years. Every product that ships AI features without thoughtful design is creating a trust debt they will repay later, usually after a public failure.