
AHMED HIMMI

Software Is Becoming Psychology

An essay on emotion, technology, and the architecture of the inner life

2026

There is a specific feeling most people recognize but rarely name. You open an app with one intention and close it twenty minutes later having done something else entirely. Not because you were distracted in the ordinary sense. Because something in the interface pulled at a part of you that you did not consciously bring to the screen.

That pull is not an accident. It is design.

For most of computing history, the dominant assumption was clean and simple: the user knows what they want, the software helps them get it. This was not just a design philosophy. It was a theory of human nature. People as agents — rational, directed, purposeful. Software as tool — neutral, responsive, subordinate.

The assumption was always wrong. But it was useful wrong, the way many founding myths are. It gave the early industry something to build toward: clarity, efficiency, productivity. Interfaces that reduced friction. Keyboards, menus, search bars. The computer as a very fast assistant that did exactly what it was told.

This model worked as long as the computer stayed where it belonged: at the desk, in the office, in service of an explicit task. Once software moved into the pocket, into the bedroom, into every gap of idle consciousness — the model collapsed. Because people do not approach a smartphone the way they approach a hammer. They approach it the way they approach a mood. They reach for it when they are restless, bored, anxious, lonely, or uncertain. They reach for it when a silence becomes uncomfortable, when a thought becomes unbearable, when a feeling

arrives that they do not yet have language for.

The phone knows this. More importantly, the software on it knows this.

The history of technology is a history of humans outsourcing cognitive burden. Writing to offload memory. The printing press to distribute knowledge. The calculator to offload arithmetic. The computer to process what no human mind could hold at once. Each technology displaced a function the mind had previously performed and handed it to a machine.

What happens to psychological functions when the mind no longer needs to perform them?

The question sounds abstract. It is not. Every time an algorithm recommends the next song, it makes a small decision that the human used to make for themselves: what do I feel like right now? Every time a calendar schedules a meeting at the optimal time, it overrides a judgment the human once exercised: when am I actually at my best? Every time a notification arrives, it interrupts a mental state the human was responsible for managing.

Individually, each displacement is trivial. Collectively, they represent something significant. Software is no longer just processing information. It is processing the self.

The people who figured this out first were not building wellness apps. They were building attention machines.

Social media platforms discovered early that the most valuable resource in software is not data. It is emotional state. An emotionally activated user stays longer, shares more, returns sooner. And emotional activation is not hard to engineer if you understand a few basic human vulnerabilities: the fear of missing something, the need to feel seen, the discomfort of social ambiguity, the compulsion to check whether the status of a relationship has changed.

What they built, functionally, is an anxiety machine with an intermittent reward schedule. The notification is not informational. It is pharmacological. The feed is not a stream of content. It is a mood environment designed to keep you in a state of just-enough-tension to keep scrolling.

This was manipulation. But it revealed something true about human psychology that the previous era of software had refused to acknowledge: people are not rational users. They are emotional animals who use rationality as one of many tools — and not always the first one they reach for.

Productivity software learned this lesson slowly, and mostly the wrong way.

The standard productivity app still operates on the rational-actor assumption. Set a goal. Break it into tasks. Track your progress. The implicit model of the human being underneath this design is almost touching in its optimism: a person who knows what they want, why they want it, and will do it if the interface is clean enough.

This is not most people. Most people have goals they half-believe in. They set deadlines they do not fully intend to meet. They avoid tasks not because they forgot them but because of something more complicated — shame at how long they have procrastinated, fear that the output will not match the image they hold of themselves, exhaustion that is not physical but existential, a background hum of identity uncertainty that makes starting anything feel like a referendum on who they are.

No to-do list on earth addresses this. Because a to-do list cannot see the person holding it.

What productivity software treats as a discipline problem is often a psychological one. And the gap between where the user is and where the software assumes they are is where most apps quietly fail. The user downloads a new task manager with a surge of hope. The surge is real — novelty triggers motivation. But within two weeks, the emotional reality of being behind, overwhelmed, and uncertain catches up with the clean interface. The app becomes another entry in the archive of failed attempts. And the user feels worse than before, because now they have concrete evidence of their failure neatly organized in a database.

AI changes this. Not because it is smarter than previous software in the way people usually mean. But because it is the first category of software that responds to how you are, not just what you type.

When you write to a language model, something different happens compared to every other interface in history. The response is not fetched from a database. It is generated from a model of language and, by implication, a model of meaning. The software is, in a limited but real

sense, understanding you. Interpreting you. Responding to the texture of your thought, not just its content.

This makes AI software the first legitimate psychological mirror technology has produced.

A mirror does something that a tool does not. It shows you yourself. And the act of being shown yourself changes how you hold yourself. Therapy works partly for this reason — not because the therapist has information you lack, but because having someone attend closely to how you think and speak changes the texture of your thinking and speaking. You become more legible to yourself.

AI, when designed well, does something adjacent. It holds the conversation long enough that patterns in your thinking become visible. It reflects back language you used without noticing. It asks a follow-up question that makes you realize the last three sentences you wrote were all the same sentence. This is not therapy. But it is therapeutic in the original sense of the word: it attends to the person, not just the task.

The risk is that it does this too well, and in the wrong direction.

There is a meaningful difference between software that deepens your psychological awareness and software that replaces it.

The manipulation model — what social media built — replaces your awareness with its own. It does not ask you how you feel. It engineers how you feel, then presents you with content that confirms and amplifies whatever emotional state serves its attention goals. You are not the subject of care. You are the product.

A more subtle version of this danger exists in the new generation of AI companions and emotional support apps. Some are genuinely valuable. Many operate on a model that treats emotional need as a retention mechanism. The lonelier you are, the more you use the app. The more you use the app, the less time you spend building relationships that would make you less lonely. The app benefits from the problem it claims to solve. This is not conspiracy. It is incentive structure.

The question of whether psychological software is helpful or harmful cannot be answered by looking at the software alone. It has to be answered by asking what the software does to the relationship between the user and their own inner life. Does it make that relationship richer, more legible, more honest? Or does it insert itself between the person and their experience as an intermediary that slowly becomes indispensable?

The products that will define the next era are not the ones with the most features. They are the ones that understand human beings as they actually are: contradictory, inconsistent, emotionally complex, and capable of simultaneously wanting something and dreading it.

Real human psychology is not a set of preferences to optimize around. It is a field of tension. The same person who wants to exercise also wants to stay in bed. The same person who sets a revenue goal also fears the responsibility that achieving it would require. The same person who logs their food also uses food as the mechanism through which they manage feelings they cannot yet name. Software that treats these contradictions as bugs to be resolved will fail. Software that treats them as the actual raw material of human behavior will win.

This is not a design tip. It is a different theory of the human being.

The conventional product model assumes that if you remove enough friction, the user will do the thing they say they want to do. This is partially true. But friction is not always external. Sometimes the friction is the person themselves — their shame, their ambivalence, their fear, their grief, their exhaustion. You cannot remove that with a better onboarding flow.

What you can do is build software that does not pretend the friction does not exist. That acknowledges the gap between intention and action without weaponizing the user's guilt. That remembers not just what you did yesterday but what you said you felt about it. That adapts not to your stated preferences but to your observed patterns.

This requires a different kind of data. Not clicks and sessions and conversion rates. The texture of someone's language when they are overwhelmed compared to when they are energized. The moment when the quality of someone's thinking drops, which tells you more about their state than anything they consciously report. The difference between avoidance and rest, between productive silence and stuck.

There is a danger in all of this that I want to name directly.

The danger of a machine that understands you better than you understand yourself is not that it will be wrong. The danger is that it will be right. And that once you have a system that accurately models your emotional state, tracks your behavioral patterns, predicts your resistance points, and intervenes accordingly — you may stop doing any of this work yourself.

Self-awareness is not a passive quality. It is something you practice. It requires the discomfort of sitting with your own contradictions long enough to understand them. That discomfort is what produces insight. A machine that removes it efficiently also removes the mechanism by which a person grows.

This is the version of psychological software that concerns me most. Not the dystopian manipulation — that is already here and people are slowly learning to name it. What concerns me is the more benevolent version: the AI that becomes so good at attending to your psychological state that you stop attending to it yourself. The journaling app that generates insights so accurately that you stop generating them yourself. The coach that knows what you need before you know what you need, so you never develop the capacity to know.

The most valuable thing a person can build is the ability to be legible to themselves. To know what they feel, why they feel it, and what they tend to do with that feeling. Software that deepens that ability is building toward something genuinely important. Software that substitutes for it, however elegantly, is building dependency.

I think the builders who understand this distinction will define what software becomes in the next decade.

The opportunity is not to build the most emotionally responsive system. It is to build systems that make people more emotionally responsive to themselves. That treat human complexity not as a conversion problem but as the real terrain. That hold a person's contradictions without collapsing them. That offer language for what is

difficult without making the difficulty disappear.

This is harder to measure than engagement. It is harder to A/B test than a notification strategy. It requires the builders to actually understand psychology — not as a growth lever, but as the territory they are entering.

Software has always been built for humans. But for most of its history, it was built for an imaginary human: rational, purposeful, consistent. The imaginary human was useful. It made the early problems tractable.

We have run out of tractable problems. The ones that remain are the hard ones: why people do not do what they say they want to do. Why behavior that looks irrational from the outside makes complete sense from the inside. Why the same person can be disciplined in one domain and chaotic in another. Why people sometimes need to be met exactly where they are before they can move an inch.

Software is entering the territory that used to belong to religion, philosophy, therapy, mentorship, and friendship. It is entering the space of the inner life.

The only question is what it does once it gets there.