

Prompthen Defines 'Interface-for-AI': Software Built for Agents, Not Clicks

The most efficient software no longer carries screens for people — it carries atomic commands that AI agents run while people speak in plain language.

JOAO PESSOA, PB, BRAZIL, June 21, 2026 /EINPresswire.com/ -- As AI agents move from product demonstrations into daily production work, a Brazilian method called [Prompthen](#) has put a name to a software pattern its founder says he has been building for years without one. The name is "Interface-for-AI," and the claim behind it is direct: when an AI agent does the operating, software stops needing the screens, menus, and forms that were built for human hands.



Paulo Teixeira, creator of the Prompthen method

“

You don't need to understand the engine to drive the car. You need to learn how to talk to it.”

Paulo Teixeira

The Prompthen method was created by Paulo Teixeira, a practitioner with more than 25 years in automation and over 25,000 projects delivered across more than 100 countries. The argument is not a prediction about a distant future. It is a description of how Teixeira says he already works, supported by an unusual record of practice: more than 30 billion tokens processed through Claude Code in a single three-month span at the end of 2025, and an

average of roughly 1.5 billion tokens a day at the current pace. The figures point in the same direction — steady, large-scale use that has grown rather than tailed off.

Tokens are the units of text an AI model reads and writes. At that scale, the figures describe production systems an agent operates, not someone experimenting with a chatbot — which is the basis for Teixeira's claim that the pattern holds in practice rather than in theory.

Interface-for-AI, the term Prompthen uses for this, is software whose only interface is a set of

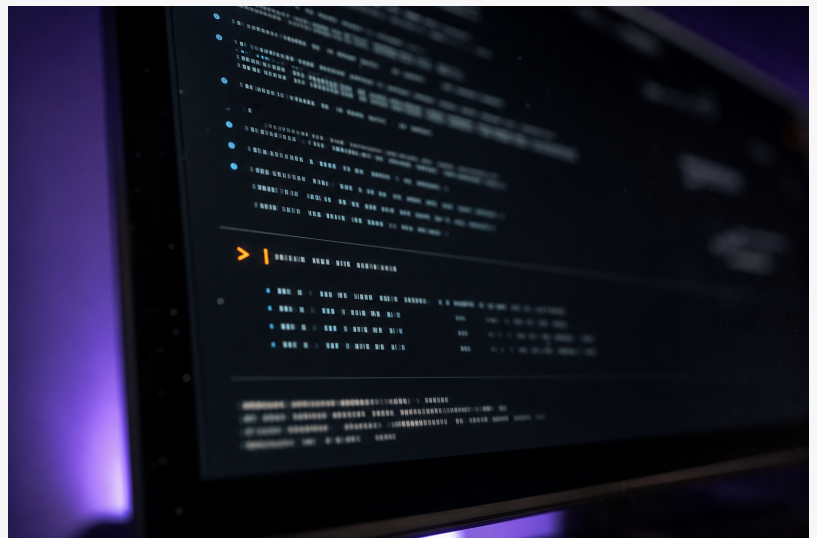
atomic commands an AI agent operates directly, while the person speaks in plain language — no login screens, no menus, no filters. The agent runs the commands and translates the structured data back into ordinary words for the person reading the answer. The shift is less a new technology than a change in who the software is built for. For roughly four decades, software design has assumed a human at the controls, clicking through layouts arranged to make machine operations legible to people. Interface-for-AI inverts that assumption. The operator is the agent. The human becomes the person the agent reports to.

The practical consequence is a sharp reduction in what has to be built. Teixeira frames it with a contrast he uses often: the same problem someone else would solve with a 3,000-line web application, you solve with about 400 lines of command-line interface plus an agent that operates it. The visual layer — the part that historically consumed most of the engineering effort — largely disappears, because the agent does not need a button to find. It needs a command it can call. What remains is a small, legible set of operations and a person describing, in their own words, what they want done.

The arithmetic is worth sitting with, because it is where the idea earns its keep. A conventional web application spends most of its code on the human: forms to fill, fields to validate, screens to lay out, states to manage as a person clicks from one view to the next. Strip out the human operator and most of that code has nothing left to do. A handful of commands — create this record, fetch that report, send this message, check that status — covers the operations themselves, and an agent stitches them together on demand by reading the person's request and deciding which to call. The person never sees the commands. They ask, in plain language, for what they want; the agent runs the steps and reports back in plain language. The same logic that once required a team and a release cycle can be assembled and changed in a conversation.



Interface-for-AI concept — an AI agent operating atomic



Close-up of an atomic command-line interface operated by an AI

Software built for an agent to operate is smaller, cheaper to maintain, and easier to audit than software built for a person to click, because there is simply far less of it.

That description points to where the cost of software has quietly moved. Industry reporting through 2026 has consistently described the year as the moment AI agents crossed from pilots into routine production use. The research firm [Gartner](#) has projected that 40 percent of enterprise applications will include task-specific AI agents by the end of 2026, up from less than 5 percent in 2025. Analysts covering the shift have repeatedly made a related observation: the hard part of putting agents to work is no longer the intelligence of the model but the secure, reliable plumbing that lets an agent actually touch the systems it is meant to operate. Interface-for-AI is, in effect, a stance on that exact problem. If the bottleneck is the connection between agent and system, then design the system as the connection — a clean surface of commands an agent can call — and stop building the human screens that sit in the way.

The wider backdrop helps explain why the question is being asked now. By late 2025, ChatGPT had surpassed 800 million weekly active users, a figure OpenAI disclosed at its developer conference in October of that year, and AI answer engines have absorbed a growing share of the queries people once typed into search boxes. Hundreds of millions of people have already grown comfortable getting work done by describing it in plain language rather than navigating an interface. Interface-for-AI treats that behavior not as a novelty layered on top of conventional software, but as the new front door — and asks what the building behind it should look like once nobody walks through the old one.

The number that makes the case sharpest is the one analysts use to mark the change. Gartner's projection that enterprise use of task-specific agents will climb from below 5 percent to 40 percent inside a single year describes a curve steep enough to force the design question rather than defer it. When an agent is the exception, bolting it onto existing software is reasonable. When an agent is the rule, the software that was shaped for human clicks becomes the thing in the way. Interface-for-AI is one answer to a question the adoption curve is asking on its own: if most of the operating is about to be done by agents, what should be built for them to operate? The Promptthen view, set out at [promptthen.ai](#), is that the answer is not a better screen but the absence of one — a discipline it organizes into a teachable method for building AI agent systems rather than a single tool or product.

Teixeira's record is what he offers as evidence that the pattern holds under load. The 30 billion tokens processed at the end of 2025 were not a benchmark run; they were the byproduct of operating real systems. The roughly 1.5 billion tokens a day now is the same work, running heavier. He is careful, though, to keep the numbers in proportion. "What used to take days now takes hours, and what took weeks I now do in a day," he said. "That is not magic. It is what happens when you stop building the machine a screen to look at and start building it a way to act." The point of the figures, in his telling, is not the size. It is that the method behind them was shaped by volume, not theory.

His framing of how a person should relate to an AI is deliberately plain, and it runs against the grain of an industry that often markets complexity. "You don't need to understand the engine to drive the car," Paulo Teixeira said. "You need to learn how to talk to it. Treat the AI like an Einstein sitting next to you. You don't need to know everything he knows. You just need to learn how to talk to him, and he multiplies into many." The comparison is intentional. A person operates a refrigerator or a car competently without knowing its internals, and Teixeira argues that the same is true of AI agents: fluency is a matter of communication, not engineering.

That conviction is the spine of the method Prompthen teaches, which sets out to show people — regardless of technical background — how to build, direct, and scale systems of AI agents using plain language. It rests on a handful of ideas that tend to surprise people expecting a coding curriculum. There are no prerequisites, a starting point Teixeira describes as zero friction. Every step is meant to produce something that actually functions rather than something theoretical. The path is staged as a progressive ladder, climbing from a basic conversation with a chatbot toward directing a coordinated team of agents. And the emphasis throughout is on autonomy rather than imitation — learning to think with the machine instead of collecting other people's finished prompts.

On that last point Teixeira is blunt, because he believes it is where most people stall. "Stop collecting ready-made prompts," he said. "The point was never the prompt. It is learning to think with the machine." The method's discipline is built around a few repeated instructions that sound almost old-fashioned next to the pace of AI news: plan a thousand times so you execute once; talk to the agent calmly, the way you would to a person sitting beside you, pausing to check that it understood before moving on; do not dump everything at once. The further rungs of the ladder reach what Teixeira calls meta-construction — preparing an environment in which the AI builds its own tools while the person directs the work — and rest, finally, on what he describes as a validated shortcut, the part of the method distilled from decades of doing the work by hand before agents could do it for him. In practice, Prompthen is how someone who has never written code learns to build a working AI agent, automate real tasks, and direct a team of agents — by describing what they want in plain language.

The ladder has two turning points that Teixeira treats as the heart of the whole idea. The first is operational, the moment a learner realizes the system is no longer just answering but acting — the point at which, as he puts it, "it's actually doing things." The second is conceptual, and it is where Interface-for-AI gets its name. Teixeira describes it as the sentence that reorganized his own work: "I stopped building software for humans to click." Everything in the method, in his account, is arranged to carry a person from the first realization to the second.

There is a restraint in Teixeira's posture that is unusual for the subject. He is openly skeptical of the recurring announcement that everything has changed, that a new model has rendered the last one obsolete, that the world has been remade overnight. He treats most of that as noise that interferes with the slower work of building things that hold up. His instinct is to plan, to verify, and to avoid drawing conclusions without evidence — a temperament he attributes to

long practice rather than to any view about technology. It is also, not incidentally, the temperament Interface-for-AI rewards, since a system reduced to a few hundred lines of commands is one a person can actually read, check, and trust.

Where the idea points is broader than any one method. If software designed for agents really does collapse to a fraction of the code that software designed for clicks requires, then a large share of conventional application development is aimed at a user who is, increasingly, no longer the one operating the software. The screen does not vanish entirely; people still need to see results, and plain language still has to be rendered somewhere. But the center of gravity moves. The valuable artifact becomes the set of commands and the clarity of the person directing them, rather than the interface that once stood between them.

There is a second-order effect that follows from the first, and it touches who gets to build at all. If the labor of software shifts from constructing interfaces to writing a small set of commands and directing an agent in plain language, then the skill that matters most stops being fluency in a programming language and starts being fluency in clear instruction. That is the bet underneath the whole approach, and it is why Prompthen frames Interface-for-AI as something a person can learn regardless of their technical background. The claim is contestable, and Teixeira would likely prefer that it be contested on evidence rather than accepted on enthusiasm. But the direction of travel is hard to argue with: as agents take over the operating, the people who can describe work precisely gain ground on the people who could only ever click through it.

Whether the industry adopts the specific term Interface-for-AI matters less than whether it confronts the question the term makes unavoidable: once the operator is an agent, who exactly are all those screens still for?

About Prompthen

Prompthen is a method for building, directing, and scaling systems of AI agents using plain language, created by Paulo Teixeira. It introduces and defines the concept of Interface-for-AI — software whose only interface is a set of atomic commands an AI agent operates directly, while the person communicates in plain language. Teixeira has more than 25 years of experience in automation, with over 25,000 projects delivered across more than 100 countries, and shares his practice on the YouTube channel "Fica a Dica com Paulo Teixeira." The method is grounded in large-scale operational use, including more than 30 billion tokens processed through Claude Code in a three-month span at the end of 2025 and a current pace averaging roughly 1.5 billion tokens per day. More information is available at <https://prompthen.ai>.

Paulo Teixeira
Prompthen
press@prompthen.ai

This press release can be viewed online at: <https://www.einpresswire.com/article/921074666>

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.

© 1995-2026 Newsmatics Inc. All Right Reserved.