

The End of Software: Why Code No Longer Matters and Relationships Are Everything

A manifesto, October 2026.

These days, isn't software the least valuable thing a software company owns?

I think software as we know it is coming to an end. Anyone can now build it, copy it or replace it, and whatever anyone can build stops protecting the company that sells it. What's left is whatever you can't download, which means well-structured data and genuinely complex solutions, and above all relationships with people.

I'm writing from daily experience (I built 80% of the software I use myself, and I don't know how to code), so I'll start with why code no longer protects anyone, then the gold rush that almost everyone building AI apps is reliving, and what software has to look like once an agent can do the work for you. After that come the one thing that can't be copied, the people companies are going to need, and the industries worth fighting in.

Code no longer protects anyone

The software I've built includes a program that manages the terminals where I run my AI agents, an AI-powered native Mac app that labels my email, answers it and sets up meetings, and a program that pulls data straight from Nasdaq and analyzes entire options chains, Greeks included, to find the best contracts and tell me whether they're actually worth trading. I used to pay a subscription for every one of those, and now I build them myself.

If someone like me, with no programming background, can build those tools, polish them to my liking and get them running, don't you think any company could? And what happens when every company embraces AI and nobody outsources software anymore? Plenty of companies will drop HubSpot the day 300 extra leads push its account into a higher contact tier and a bigger monthly bill.

The markets have already noticed. As I write this, in mid-October 2026, HubSpot is down 42% year to date and Monday.com is down 39%, while Shopify was down 41% at one point this spring. The selloff even has a name now, the SaaSpocalypse, which is shorthand for the fear that AI agents able to do the work on their own will make per-seat licensing obsolete.

That makes sense to me, given that there's an open-source option for almost everything and cheap AI models that answer questions and write code. Software has been losing value layer by layer: first the frontend (now very easy to build), then the backend, and finally infrastructure and operations. Now add REA (Reverse Engineer Anything), an open-source tool that lets an AI agent reverse-engineer any app installed on your computer, even one from a company worth hundreds of millions of dollars, and then build you your own version. Small businesses are already asking the obvious question: why pay hundreds of euros a month for third-party software when a tool like that exists? Code doesn't work as a moat anymore, and generic SaaS has gone down with it.

Everyone is digging for gold, and only the shovel sellers win

And yet almost everyone keeps building software expecting to sell it. I was stuck in that trap for months, seeing AI as a way to supercharge my ideas and build solutions for companies faster. All I was really doing was building traditional solutions with AI instead of accepting that we don't have to keep building the same thing.

Right now the default move is to pay \$200 a month for an AI coding tool and use it to build general-purpose software that anyone can adapt. The result looks exactly like fast fashion, a market drowning in supply, full of sloppy, disposable products. I think there's a bubble in business AI apps that all look alike, with nothing innovative about them, trying to sell a solution today that would have been fantastic six years ago. Lots of people are building, very few are making it and almost nobody is selling, because when everything is this easy the only thing left is a price war. The moment a rival ships a feature, markets it better or is simply cheaper, you have to match it, and it gets so competitive that you can't grow. You stop thinking about growing and start thinking about holding on, and that's a war I won't fight.

It reminds me of the California gold rush, when people came from all over the world to strike it rich, and in the end the only ones who got rich were the ones selling shovels. Today the shovels are the AI models that write code, the data centers and the chips. Everyone knows that part. For anyone building software, though, it means that every \$200 monthly subscription someone pays to build generic software and sell it is a shovel sold, and the seller

gets paid every time, whether the digger strikes gold or not. And the gold, meaning generic software, is now worthless, because everyone is digging with the same shovel and pulling out the same thing. What I mean is that you shouldn't be just another sheep in the herd, building software for nobody. Or for everybody. I think it's idiotic to start a company building an agent that does everything, because an agent built to do everything ends up doing nothing well.

Why five apps and not one?

Take my own case.

I don't like Outlook, Teams or Calendar, but I need them every day, and inside them I only do a handful of things: send emails, save drafts, schedule sends and book meetings. Why would I do that in an app with 300 features? The logical move would be some version of the Pareto principle, keeping the 20% of features that handle 80% of the work. That's how most people are approaching it, and on top of that they bolt an agent onto a panel on the right. Great.

We keep working like this only because that's how it's always been done. I'm not talking about Microsoft or HubSpot in particular; I'm talking about SaaS in general. If after years on a platform you can't improve it, or you can't think of a single way to make it fit you better, you haven't thought about it enough. SaaS is built for the masses so that it works for everyone, but why should we use a system where we have to click around and spend hours managing it? Can't we just forget traditional SaaS and embrace the new era?

In that new era, the agent works inside the app itself. It can perform every action the app offers, it moves between screens, it understands context and it writes like you, because while you were building it you gave it all your emails so it could learn your style. If you can build that, why would you treat AI as an add-on rather than the core of the product? Why can't you just ask it what to do, how to do it or what to reply, instead of digging through your inbox?

The agent and the interface draw on the same data. This agent oversees the work and carries it out, and it can do anything the app can do, because it works directly on the app's data and actions rather than on buttons someone designed for everybody. You teach it and give it rules to follow. For example, I told mine that whenever it proposes two dates in an email, it should create a calendar invite for each one with "(TC)" at the start of the title, for "to confirm". That way, while I'm waiting for the client to tell me which slot works, I don't book another meeting on top of it and then have to tell them the slot is gone, which makes me look bad. You can ask it to alert you about anything at all, not just whatever a list of preset options allows, and every

time you correct it, it learns. Because it knows the whole app, when you ask for something that doesn't exist yet, it puts it together from the actions it already knows, so the agent ends up building itself around you. Why fight an interface and features you don't care about when it can do that for you? I made this point in my essay "La máquina del futuro" (The Machine of the Future): the machine of the future is the tool that can take action in the real world simply from a plain-language description of what you need.

That's why I'm proposing a new way of working, where the interface stops being where you act and becomes a live view of what's happening. The agent acts, and I sit next to it, watching and correcting it as it goes. Email and the stock market aren't going anywhere in the next decade, and neither is the habit of running AI alongside them.

All these new AI companies want to take us back to the same old thing. On a typical day I already juggle four different apps, and every one of these new AI companies wants to be the fifth. And why do I need five in the first place? Why not two, or just one of my own that handles everything while I keep an eye on how it behaves and spend my time on more interesting, more creative things? We have to stop building 2010-era software and start building connected, intelligent software that lives outside conventional apps.

The only thing that can't be copied

If code doesn't protect anyone anymore, what does? A small company will build its own software with its own AI specialist. A big one, before it builds anything, needs data infrastructure across the entire company. That's why I've focused on what I think is hardest to replace: very complex solutions for very large companies. It sounds obvious, even easy to replicate, until you realize that a single solution means integrating with 15 different APIs, setting up your own databases and building flawless data pipelines so the AI doesn't hallucinate. AI writes the code for all of that too, these days. What no model can hand you is an understanding of a large company's data, or the trust it takes for that company to open its data up to you. The fact that someone who can't code built all of my own software is proof enough that code isn't the hard part anymore.

That's where the human relationship comes in, and I don't see it going anywhere. In that same essay I wrote that a company's value lies in the people who work there, and always will, however far technology advances. I think software will become increasingly custom-built, developed together with the client company, with outside partners working inside the business to solve its problems. Unless an idea is truly disruptive, I don't think generic

software has much of a future, because the market for generic tools is already saturated with people and companies selling software. A company hands its data to someone it already has a relationship with. That's why I say relationships are everything: anyone can copy a vendor's code, but the relationship stays with whoever knows the client's data and problems.

So I believe the right order is to build the company's own data infrastructure first, across the whole company and with proper data governance, and only then build specific apps for the company or for one department on top of it. Whatever progress we make as a society will come from the average company having better-structured, more complete data, much more than from having more tools. Instead of trying to build better CRMs or ERPs, like everyone else, we need to build solutions that make sure a company never needs a CRM again, with specific apps for specific employees.

Who wants to hire a HubSpot expert, pay thousands of euros for a new employee and, on top of that, keep paying thousands more every month for a platform that sits on your data and locks you into its services? When the system is yours, your data can live on your own machines in your own office, and you don't have to worry about where it ends up. Isn't getting out of that one of the greatest advances in history?

Every company's Avengers

Companies will gradually stop adapting to generic software that someone else built for everyone, and they'll start building their own, because the barrier to entry is minimal. In small companies, one prompt engineer will do all of that work, and in large ones there will be one in every department, responsible for every pain point their colleagues have. They'll be the company's Avengers, the people who modernize it and do the work that actually moves it forward, and they'll put an end to SEO, to junior roles and to the high cost of paying people for repetitive tasks. That doesn't mean you stop hiring; you'll hire people for how they think and for their potential to grow inside the company, not for being better at a mechanical task, since a system trained for that can already handle it.

I know plenty of companies still resist change, or haven't been exposed to AI much, and see it as a chatbot that answers questions. They don't see everything it can do when it oversees and carries out the work of an entire department. But those same companies are swamped with day-to-day work, and they'll be increasingly open to hiring real experts who can handle their data, because their Avengers can't build anything on top of data nobody has organized. I'm just a little ahead of the curve because I use AI heavily every single day.

No pirates and no price wars

If relationships are everything, it matters a great deal who you have them with. Some industries have far more complex problems than others, and plenty are much better bets than marketing.

I know this is going to annoy people, but I see marketing agencies as pirates. They take other people's services and resell them to the end client, adding management or content on top. They don't actually make anything (the only thing they really market is themselves). They're ruthless on price, because their whole business is margins and client management. They'll fight over every last euro, and they'll always go after your margins, because that's exactly their job: negotiating on both sides of the table. I've seen this firsthand at other companies, and while not every agency is like this, most of the ones I've dealt with are loyal to no one. They set false expectations that someone else has to deliver on, they're extremely demanding and they're rarely grateful. They're masters of reselling and negotiating, but that doesn't mean you have to do business with them.

They take that pressure from their own clients and pass it down the food chain, and whoever builds software for marketing walks straight into the middle of it. With this much competition, the whole sector has turned into a battle of pirates. And trust me, you wouldn't want to be on anyone's ship.

Competition leads to mediocrity rather than success, which is why I prefer industries like healthcare, defense or transportation, where the problems are more serious, where you don't compete on price, and where you can grow as far as you need to without always having a market leader to beat. In those industries, keeping your data on your own machines, where it never leaves your hands, matters even more. As Peter Thiel put it, "Competition is for losers."

Software adapts to you, not the other way around

No more adapting to software, buying licenses and learning how to use it. If there's one thing I want people to take away from all this, it's that software isn't worth anything anymore and relationships are everything.

Sources

Share prices: Yahoo Finance ([HubSpot](#), [Monday.com](#), [Shopify](#)); the year-to-date change is the close of October 16, 2026, against the close of December 31, 2025, and Shopify's figure is its low of May 2026. SaaS apocalypse: Peter Cohan, [Forbes, February 6, 2026](#). HubSpot contact tiers: [HubSpot Knowledge Base, "Set contacts as marketing"](#). REA: github.com/morluto/rea. Peter Thiel, ["Competition Is for Losers", The Wall Street Journal, 2014](#). The essay quoted is [The machine of the future](#).