FINAL VERSION

What If the Most Important Thing We Teach Our Kids Has Nothing to Do With AI?

You already know what you actually want for your kids.

Not a job title. Not a salary. Not a credential hanging on a wall. Something deeper than any of those — capability, confidence, a sense of purpose, a life that feels worth living. A kid who grows into an adult who wakes up in the morning and knows why they're getting out of bed.

That's the real ask. And if you're honest with yourself, it's the only one that keeps you up at night.

So before we talk about artificial intelligence — before we talk about tools, platforms, models, or any of the breathless predictions filling your feed — let's talk about something that has been true for every generation that ever lived, and is more urgent right now than it has ever been.

There is a skill that turns curiosity into capability. Capability into confidence. Confidence into contribution. Contribution into identity. And identity, over time, into a life that feels genuinely satisfying.

That skill is systems thinking.

And here's the thing almost nobody tells you: you already do it. You've done it your whole life. You just never had a name for it.

The Framework

Every time a human being sets out to accomplish something they've never done before, the same pattern emerges. It doesn't matter whether they're sixteen or sixty. Whether they're fixing something, building something, cooking something, or learning something. The pattern is always the same.

First, there's an objective. Not a vague wish — a real one. Something you're trying to complete, with a clear picture of what done looks like. Not "I want to get better at cooking." Something more like: *I want to put a smoked brisket on the table at Thanksgiving that makes people put down their forks and stare at the ceiling.*

Second, there's an environment. Every objective lives inside a system. A context. A world with its own rules, vocabulary, tools, and culture. Before you can operate effectively inside that world, you have to understand it well enough to ask the right questions. You don't have to become an expert. You have to become oriented.

Here's what that feels like in practice: you're standing at the edge of something you've never done. The terminology is foreign. The tools are unfamiliar. The unwritten rules haven't been explained. You don't even know what you don't know yet. That moment — right there, before you touch anything — is where systems thinking begins. Not with action. With orientation.

Third, there are tools. Once you understand the environment, you can identify what you actually need to do the work. Not before. The tools only make sense once the environment makes sense.

Fourth, you execute. You go. You build the thing, cook the thing, fix the thing, swim the race.

And then — this is the part nobody warns you about, and the part that changes everything — **you do it again.**

Because the jump from the first time to the second time is the biggest leap you will ever make. Bigger than any jump after it. The first time teaches you what you didn't know you didn't know. The second time, you actually know where you are. Every attempt after that gets better, but nothing matches that first-to-second transformation.

Expect it. Accept it. Make it part of the deal you make with yourself before you begin.

This is systems thinking. Four steps, one iteration principle, applicable to anything.

And AI — used correctly — is the resource that lets you learn the environment faster than any previous generation could. Not a replacement for the process. An accelerant for the learning.



The Swimming Civilization

My daughter Brittney is an eight-time All-American swimmer. Four-time state champion. Conference champion in college. Full athletic scholarship. She is, by any measure, someone who figured out how to compete at the highest levels of her sport.

When she was seven years old, my wife Suzanne and I knew absolutely nothing about competitive swimming.



Not a single thing.

We didn't know the difference between a 50-meter pool and a 50-yard pool — and it turns out that difference matters enormously, because world records are set in one and not the other, and your child's qualifying times are calculated differently depending on which one they're swimming in. We didn't know what a flip turn was, or why it mattered, or what it felt like to push off a wall at full speed underwater and come up already moving. We didn't know about starting blocks, or the entirely different technique required for an in-water start.

We didn't know that backstroke swimmers navigate by flags strung across the pool — a system built so they know exactly how many strokes remain before the wall, so they don't crack their skull into the concrete at full speed. We didn't know about goggles and caps — specifically, that competitive swimmers often wear two caps, and that the goggle strap goes between them, not on top, because it reduces drag by a fraction of a second that turns out to matter enormously.

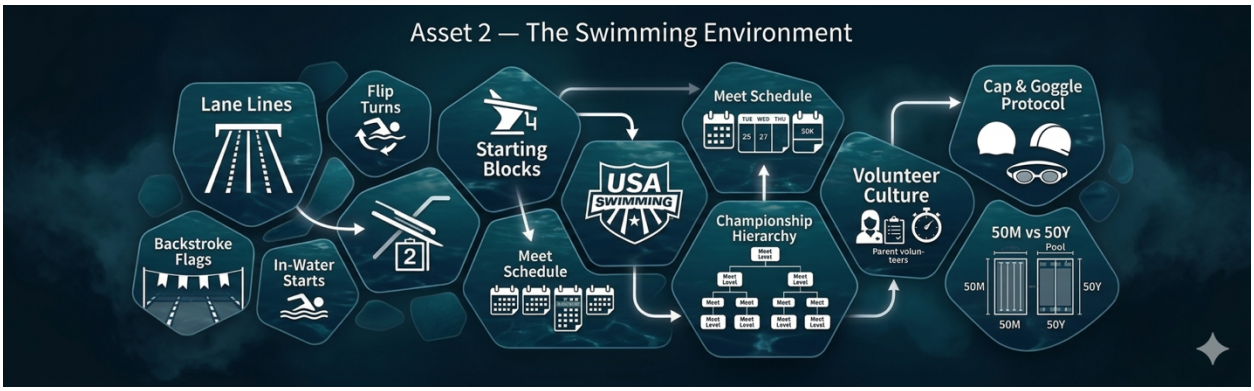
We didn't know about the national governing body. We didn't know that there was an entire organizational structure — USA Swimming — that governed everything from local club meets to Olympic Trials. We didn't know about meet schedules, championship hierarchies, cut times, qualifying standards, or the difference between a C meet and a championship final.

We didn't know that parents were expected to volunteer. To time. To judge. To work the concession stand. That the club didn't run without the families showing up and doing the work.

We didn't know any of it.

What we had was a seven-year-old who liked the water and two parents willing to learn.

So we learned. Meet by meet, season by season, coach by coach. Andy taught us one layer. Grant taught us another. Wolfgang, Jim, and Bob each added something the others hadn't. No single coach held the entire map. The environment was bigger than any one teacher — and learning it required all of them, over years, through participation.



I traveled to Germany once for a training trip and watched swimmers share a pool with no lane lines at all. A free-for-all, everyone navigating by feel and instinct and awareness of the bodies around them. Same sport. Entirely different system. It crystallized something I hadn't been able to articulate before: the environment isn't the activity. It's the organized world around it. Two people who both swim can be operating in completely different civilizations.

I've thought about what would have been different if I'd had AI when we started.

I would have sat down before the first practice and asked: *explain the world of competitive youth swimming to someone who knows nothing.* And I would have gotten in twenty minutes what took us two years to piece together. I would have understood the system before we were inside it — and I would have been able to participate more effectively from day one.

Brittney still would have had to do the swimming. The work was always hers. But I would have been a better partner to her earlier. And that matters.

That's the environment step. That's systems thinking.

Pool. Governing body. Coaches. Culture. Volunteer structure. Meet hierarchy. Cap protocol. Backstroke flags. Flip turns. All of it — the environment — before you can meaningfully participate in a single event.

The Brisket



A few years ago, Suzanne and I were at a barbecue restaurant. One of those places where the brisket arrives on butcher paper and the line starts at ten in the morning. She took one bite, looked up, and said: *"Can't you learn to make one of those?"*

Reader, I could not make one of those.

I didn't know what cut of beef a brisket was. I didn't know what part of the animal it came from. I didn't know whether you cooked it in an oven, on a grill, or in something called a smoker — and I definitely didn't know what a smoker was or how one worked.

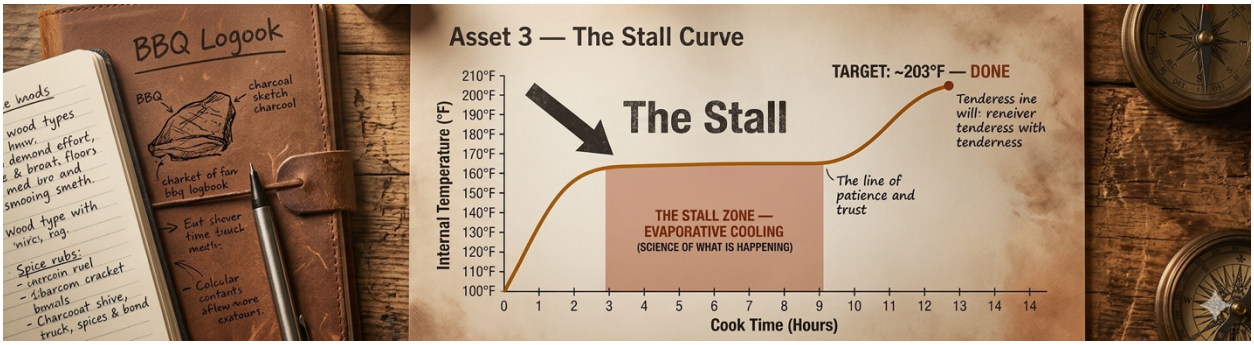
So I started at the environment.

I learned that brisket is a cut from the lower chest of a cow. That it's one of the toughest cuts on the animal — full of connective tissue that has to be broken down over a long, slow cook. That temperature is everything. That competitive pitmasters don't guess; they measure, obsessively, with wireless probes that send readings to their phones so they can monitor the cook from the couch.

I learned about the stall.

The stall is the moment — usually somewhere between 150 and 170 degrees — when the internal temperature of the brisket stops rising for hours. Just stops. A complete beginner sees this and panics. They think something has gone wrong. What's actually happening is evaporative cooling: moisture leaving the meat surface is keeping the temperature flat, like sweat cooling

your skin on a hot day. The stall is not a problem. It's the system working exactly as it should. You can push through it by wrapping the brisket in butcher paper or foil — the Texas Crutch — or you can ride it out. Both are valid. Both require knowing what the stall is and why it happens.



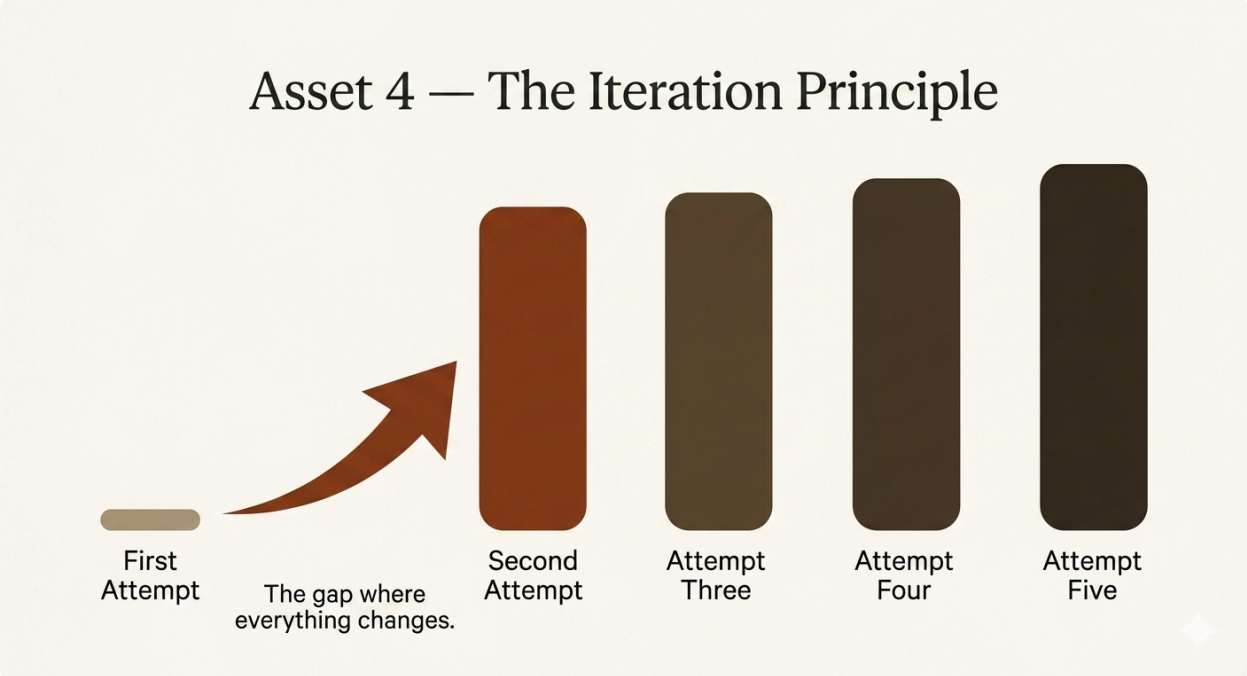
I learned about trimming. About dry rub. About injection. About smoke rings and bark and the difference between a brisket that tastes like roast beef and one that tastes like something you'd wait two hours in line for.

The tools followed from the environment: the smoker, the probes, the paper, the rub.

Then I executed.

The first one was educational. I learned what I didn't know I didn't know — which turned out to be considerable. The bark wasn't right. The slice was too thick. I pushed through the stall wrong.

The second one was meat candy. People put down their forks and stared at the ceiling.



That's the jump. First to second. Every time. The first brisket wasn't a failure — it was the tuition. It bought me the knowledge that made the second one extraordinary. And now, with AI walking you through the environment before you even light the smoker, even your first attempt starts from a higher floor than mine did.

Same framework as the swimming pool. Smaller system, identical structure. Objective, environment, tools, execution, iteration.

The Software

This one's for the parents who think the previous two examples were charming but this is where it gets real.

Thirty years ago I was building things professionally. Then life took me elsewhere — business development, sales, funding, strategy — and I stepped away from hands-on building for three decades. No programming. No development environments. No code.

Two years ago I came back.

The first wall I hit was authentication. I was building an application that needed users to log in securely, and I sat in front of my screen for an uncomfortable amount of time not knowing what I didn't know. I didn't know if authentication was something you built yourself or something you plugged in. I didn't know what OAuth was, or why it existed, or what the consequences were of getting it wrong. I didn't know the difference between a token and a session, or why that difference mattered at two in the morning when something broke in production.

So I stopped. I didn't touch the code. I asked AI to orient me: *explain the world of authentication to someone building an application for the first time. What are my options, what are the tradeoffs, and what do I need to understand before I write a single line?*

That question — asked in some form, at every threshold — is the environment step.

I learned what GCP was and why it existed. I learned why you might choose Firebase over a Cloud Run function depending on what you're building and how fast you need to move. I learned the difference between Firestore and ChromaDB and when each one makes sense. I learned what a terminal-based architecture means and when you'd reach for one. I learned the development environment, the testing framework, the debugging process, the deployment pipeline.

And then I built. Badly at first, as promised. And then dramatically better.

I built iOS apps. I built AI tools. I built systems using native vector embeddings. I built Recordari, a personal memory companion that captures and recalls life details through natural conversation. All of it from zero. All of it using exactly this framework.

The development environment. The testing setup. The debugging process. The deployment pipeline. All of it — the environment — before a single line of code.

The Pattern Underneath Everything

Pool or track. Oven or smoker. VS Code or a terminal window. The environment changes every time. The framework never does.

This is the thing worth saying plainly: you have already done this. Every time you figured out something you'd never done before — every time you walked into an unfamiliar system and eventually found your footing — you were doing this. You just didn't have a name for it. And because you didn't have a name for it, you couldn't teach it deliberately. You could only hope your kids absorbed it by watching you.

Some will push back here and say: kids today won't engage with this. They want everything handed to them. They don't want to read, research, probe, or explore. Just tell me the answer.

That's real. And it's the symptom, not the diagnosis.

Curiosity doesn't have to be the entry point. Accomplishment is.

You don't need a curious kid to start. You need a kid who finishes something — anything — and gets recognized for it. A parent who says *you figured that out.* A grandparent who takes a bite and stares at the ceiling. A coach who says *do you see what you just did?* That moment of earned recognition rewires the loop. Now they're curious — because curiosity has a payoff they've actually felt. Because the dopamine of finishing something real and being seen for it is a fundamentally different sensation than the dopamine of scrolling past someone else's highlight reel.

One makes you feel capable. The other makes you feel like you're missing something.

Our kids have drifted farther from the first feeling than any generation before them. Post-COVID. Post-social media. Forming their identities at the exact moment the most powerful technology in human history is arriving — and being told by the loudest voices in the room that the future is a place where they won't have to do anything.

AI is not a path toward having everything spoon-fed.

It is a path toward rebuilding the thing that got lost — participation. Meaningful engagement with the systems that shape your life. The ability to step into a world you don't understand and orient yourself, learn the environment, find the tools, and do something real.

Capable. Confident. Contributing. Present in their own lives. Satisfied not because everything was easy, but because they showed up, learned the ship, and steered it somewhere that mattered.

That chain hasn't changed in ten thousand years. AI didn't break it. Used correctly, it's the first tool in a long time that might help us repair it.

And that — more than any tool, any platform, any model, any prediction about what's coming — is worth teaching.

Addendum: A Note From Todd

My wife doesn't ask me to theorize about systems thinking. She asks me to make a brisket.

She also once mentioned — casually, the way she mentions things that turn into month-long projects — that the car was making a grinding noise when she started it in the morning.

I had never touched a starter in my life. I wasn't sure I could identify one in a lineup. But I sat down, described the symptom to AI, and asked it to orient me: *what are the possible causes of a grinding noise on ignition, and what do I need to understand about the electrical and starter system of this specific vehicle before I start pulling things apart?*

Two hours later I understood the starter motor, the flywheel, the solenoid, and why the grinding meant the gear wasn't engaging cleanly. I sourced the part. I pulled the old one. I installed the new one in the driveway on a Saturday afternoon.

It wasn't pretty. I skinned two knuckles and said things I won't repeat here. But it worked.

Same framework. Different garage.

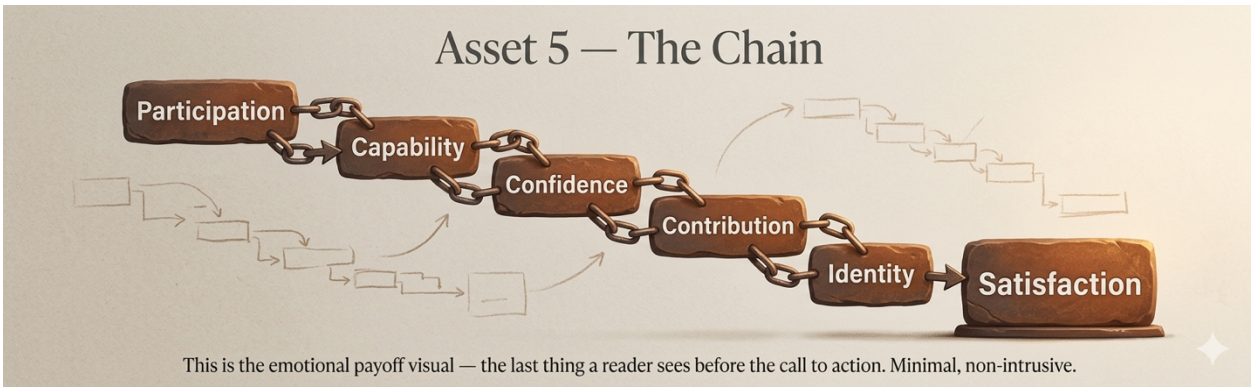
I came back to building two years ago after a thirty-year gap. No recent programming background. No CS degree. No team. Just a framework I'd been using my whole life without realizing it had a name, and a tool that could help me learn any environment I needed to enter.

I built iOS apps. I built AI tools. I built native vector embedding systems. I built Recordari. I cooked the brisket. I replaced the starter. I fixed the AC on a Saturday in August when it decided to stop working at noon.

Every single time: objective, environment, tools, execution, iteration.

The first version was always the worst. The second was always dramatically better. And somewhere in that gap — in the space between not knowing and figuring it out — something happened that no algorithm can replicate.

I felt capable. And capable, it turns out, is where everything else begins.



The Roundup

What you just read describes something that used to happen naturally.

Two hundred years ago a farm kid learned systems thinking the same way they learned everything else — by doing it alongside the people around them. Weather, crops, animals, tools, finances, community. All of it connected. All of it lived. Nobody called it systems thinking. It was just how you grew up. The transmission happened because the environment demanded it.

That environment is gone. And nothing has replaced it.

The Roundup exists because of that gap.

It is a program for families with high school and college-age children built around one conviction: the young people who thrive in an AI-infused world won't be the ones who learned the most tools. They'll be the ones who learned to think — about systems, about environments, about what they're actually trying to build and why.

What the farm did naturally for previous generations, The Roundup does deliberately for this one. It gives young people a structured path toward the kind of systems-level thinking that builds real capability, real confidence, and a life that feels worth living.

Not a course. Not a curriculum. A framework for how to participate meaningfully in a world that keeps changing — and the guidance to actually use it.

If what you read here resonated, the next step is simple.

Come see what we're building.