

Built Anyway

What I Learned Starting KAAI TECH LLC and Shipping Everything That Came After

Kory Kaai

CEO and founder, **KAAI TECH LLC** · CTO · Software developer

Sixteen years in the military: airborne infantry and medic

2026

A quick word before we begin

This book is part memoir and part field notes. I am not selling a fantasy about four-hour workweeks or passive income on a beach. I am telling you what it felt like to put **KAAI TECH LLC** on paper, wear the CEO hat, keep writing code, and put real products in people's hands while the world asked for more speed and fewer excuses.

You will read about the long work of **building real skills** and what **cybersecurity** taught me, about **service** and coming home, about **freelance** rooms and **non-technical executives**, about **contracts** and refusing to run a **charity** with a business name on the door. You will read why I believe **humans still steer AI** no matter what the feed claims, and a longer piece about a **friend, Bitcoin**, and what an **investment journey** can teach you about **trading, greed, and discipline** (still not investing advice, just what it felt like to watch from close up). You will read how **going against the grain** felt compared to staying inside someone else's ladder, and what broke, what held, and what I would hand my past self if I could.

I will not invent war stories or fake clients for drama. If you laugh a few times, good. If you close the book with one idea you can use Monday morning, even better.



Part one: From the keyboard to the letterhead



I did not slide out of a venture studio with a brand deck and a warm intro list. Technology pulled me in early. As a teenager I was already programming and pulling computers apart to see how they worked, including the usual youthful experiments with scripts around computer games. It was not always pretty, but it was honest practice in logic, patience, and testing what you think you know. I still remember getting something to work at one in the morning while the house slept, not because midnight coding is heroic, but because it proved I could bend a machine to an idea if I stayed with the problem. That hunger for closure never left. It helps when you release work. It hurts when you need to stop and sleep.

None of what came later happened because talent fell from the sky. I put in years of **deliberate work** on the craft: reading until the vocabulary stopped sounding hollow, breaking things in safe places so I could see how failure behaves, returning to fundamentals when a shiny framework made me lazy, and accepting that **competence is repetitive** long before it looks effortless from the outside. I went deep on systems, networking, and code because shallow knowledge does not hold when production is staring you down at the wrong hour. I treated skill-building like training, not a single event. Some weeks progress was invisible. Then a problem arrived that would have flattened me years earlier, and I felt how my mind moved differently.

Cybersecurity became part of that training in a serious way, not as a compliance checkbox. **Defense is layered**: one gate is not a full plan, it buys time. **Least privilege** and **secrets hygiene** matter; a credential pasted into a chat “just for a minute” has a way of lasting forever. In most places **people are the most attacked surface**, not because they are foolish, but because hurry, pressure, and trust can be exploited. “We will harden it later” is a promise incidents love, because later is when nobody remembers who said what. **Dependencies and patching, logs you can read when you are tired**, and **assuming breach** long enough to rehearse detect, contain, and recover beat making it up under adrenaline. When I started running **KAAI TECH LLC** and releasing products, those ideas stopped living on slides and became habit: how you hold customer data, how you design authentication, how you choose third parties, and how straight you are about what you cannot promise. I am still learning. The aim is not paranoia. The aim is to **build and operate like livelihoods and trust are on the line**, because they are.

Then there were **sixteen years** in the military as **airborne infantry** and **medic**. I will not turn this into a war movie. I will say plainly that part of that life is leaving a familiar version of yourself behind, learning where stakes are not theoretical, and then trying to live a normal Tuesday while the world expects you to slot back in without a seam. Going to war, in the human sense, is not only a place on a map. It is carrying what you cannot fully explain, finding what you are made of when you are past tired, and learning that **making it out** is both a moment and a long walk. The paperwork ends. The body returns. The mind sometimes follows later.

Infantry shows you the map is not the ground. Airborne reminds you that you do not pause physics, and that preparation is rehearsals and checks, not vibes. Medic work shows you that people are not deployments or “edge cases,” and that clarity matters when someone else is scared. Those years also showed me **toxic leadership** in more rooms than I wish were true, uniformed or not. Power without accountability rots the culture. The piece you control is your own example: **lead by showing how it is done**, not by shouting slogans. I decided I would rather be tired from doing the right thing quietly than clever from cutting corners unseen. **Discipline is everything** because it is how you keep promises to yourself when nobody is clapping. Later, in enterprise meetings where the energy went to naming a phase gate, part of me counted minutes we were not spending on rollback plans, and part remembered how fast people turn when leaders treat humans like inventory.

After that chapter I came up through **IT leadership, modernization**, and long **freelance** runs for **many companies**. Freelance is psychology: speed versus documentation versus “partnership” that means cheap, fast, and perfect. Over time I got faster at hearing the real constraint, whether budget, fear, or a stakeholder who needed a win. Scope creep wears a friendly mask: “while you are in there,” “just a small tweak.” Without a **written boundary**, you pay in evenings you never get back. That lesson would grow into a rule I now treat as non-negotiable for real work: **get it in writing before you bleed hours**.

The expected path is that you keep climbing someone else’s ladder until the title matches your skills. I respect that path. It was not the one I wanted to give my life to. I wanted **ownership of outcomes**, not endless alignment on other people’s roadmaps. I wanted to get work out the door without a calendar of meetings about meetings. **Starting my own business** instead of staying only inside a

company meant betting on my judgment in public, living with income that does not arrive like payroll, and accepting that some people will never understand giving up the illusion of stability for the reality of building. The naive voice said I would finally have control. The realistic voice said you swap one boss for many small bosses: customers, vendors, and your own standards. Still worth it. The house got quieter in some ways, and louder in others.

In **2017** I put **KAAI TECH LLC** on paper. Forms, fees, a PDF that makes it official. Then you sit at the keyboard and notice the world did not clear its schedule for your letterhead. I already knew I loved building things people could click. I did not yet know how much of the job would be *not* building: scope, **contracts**, support, and explaining the same risk three times to three people who heard three different things. If you have the right sense of humor, that is comedy, usually around month nine.

Part two: Year zero and the pull on your time



After **2017** the company was real on paper and the world still wanted proof.

Running the company made clear there are always **more people who want your time than hours that tolerate fantasy**. Stakeholders multiply. Everyone's fire feels unique. I started judging work by what actually helped the user and the company, not by whoever shouted last in the inbox. Some relationships were real collaboration. Others were slow extraction: they wanted **my skills** without wanting to pay for the judgment that makes those skills safe in production. **Hard lessons** stacked up: people who treat your patience as unlimited, who confuse your expertise for a favor machine.

Non-technical executives are not villains in my experience. Many are excellent at their lane. The friction hits when decisions require work they have never done. Without a shared language, technology becomes magic on one day and a cost line to squeeze the next. You end up translating, bringing options with tradeoffs, and recognizing when you are being asked to carry risk someone else does not want to understand.

The longer I sold skill for money, the more I treated paperwork as part of the craft, not an insult to trust. I stopped starting serious work on a handshake alone. A real agreement forces the boring questions early: what we are delivering, how and when we get paid, what happens when the scope shifts, who owns what when the dust settles. None of that is cynicism; it is a way to be kind to your future self, because the first time money or blame is on the table, the friendly vagueness you relied on in week one disappears fast. Buyers will often push for the lowest number that still keeps you in the

room, sometimes because their budget is real, sometimes because they are testing whether you are hungry enough to discount your own tomorrow. I had to learn, slowly and with a few bruises, that I was running a company, not a charity, and that walking away from a bad fit is not melodrama. It is how you protect the work you still owe the clients who actually value it. Every “nice” discount I gave away for free taught the market to ask for the next one.

Registering the LLC felt like a coronation for maybe forty minutes, until the first person asked what I actually did and I watched the polite clock start ticking. I settled on a pitch I could say without running out of breath: we build software products and help small teams get running on modern stacks. If they leaned in, I went deeper; if they did not, I stopped talking. I used to bury people in detail because I was nervous. Curiosity earns the second paragraph; anxiety does not. Enthusiasm without a clear point is just loud talk.

Running as CEO while you are still the one opening the error trace means you never get to float above the work in your mind, and I prefer it that way. The harder lesson was boundaries. When I answered every Slack at midnight, I was not proving dedication; I was training everyone, including myself, that midnight was normal. It took me two rounds of burnout-flavored evidence to believe that urgency is not the same as importance, and that if I treated every notification like a five-alarm fire, I would eventually sleep through the one that actually was.

Money turned out to be similarly unromantic. Cash flow does not care how clever the idea sounded in the shower. It cares whether someone paid, whether the invoice matched the work, and whether we closed the loop without turning “busy” into a trophy. I started caring more about clean finishes than about looking busy, because busy is cheap and finished is rare.

Production has its own way of taking your ego down a notch. My failures still pair the same two excuses: “we really thought we covered that edge case” and “the user did something so human we almost had to applaud.” What changed over the years was how fast I moved and how little I talked while doing it: fix first, explain second, leave a note so the same bug does not visit next month. If a log file has not humbled you yet, you have probably not released often enough. It will find you.

There is a kind of Friday afternoon that feels innocent until it is not, when people are already mentally halfway into the weekend and something in the stack decides to remind you who is really in charge. Maybe a dependency shifted, maybe a config drifted, maybe a user did exactly the thing you laughed about in standup. The version of you that survives that moment is not the one with the best hot take; it is the one who reaches for the runbook, reads the last deploy, walks the diff, and limits how much can break on purpose. I stopped treating “never deploy on Friday” as religion and started asking a simpler question instead: if I release this now, am I willing to own the weekend emotionally before I click the button? If the answer is no, I wait.

Part three: The portfolio (what stayed with me)



I will not hand you a brochure for every product. I will tell you what stuck after building them, because that is what survives.

ClientsDock came from watching smart people drown in email archaeology. Agencies and freelancers want clients to stop asking “which file was final?” They want a portal: here is the work, status, next step. If onboarding feels like homework, you lose them before value lands. The first session should feel like relief.

Erikinos (Italian learning) humbled me about pedagogy. Language products look easy in a deck until you watch real humans forget words at the worst times, which is all the time. Motivation is the rhythm of the product, not a sticker on the last screen.

Kaai.io lives in careers: resumes, tailoring, interviews. The market shouts transformation. I care about the boring win: an hour saved before a deadline, less anxiety, a tool that feels like a coach who respects your time.

Kory Grinds is habits and execution, close to the bone. Building for yourself is hard because you forgive your own inconsistency in ways you would never allow from a customer.

MutuoItalia sits in mortgages in Italy: regulated, emotional. You slow down, triple-check copy, and remember speed without care is recklessness with someone else’s life choice.

NameMetrix and domains: numbers beat adjectives; people smell hype. If I cannot point to a method, I do not play oracle.

Stack Analyzer came from fatigue with sprawl pretending to be optimization. Sometimes the product is a mirror: show what you pay for, let them decide.

TopDailyTools proves curation is work. Lists are easy. Quality over time is not. Consistency beats genius on the open web.

SparkyNexus Trainer (Pokemon battle training) is my reminder that joy is allowed and your inner twelve-year-old deserves a win.

The pattern repeats: pick a user, pick a pain, deliver something that does not need a phone-book manual. I have thrown away shower-brilliant features that died in production. Smaller slices and quicker feedback from real users beat rewriting your soul. **Selling**, for me, stopped being a costume when I reframed it as helping someone see whether I can solve their problem. If not, say so. If yes, show how. The best close I had sounded like a calm conversation: no villain music, no fake urgency on a landing page, just clarity.

Part four: The hype cycle, AI, and who is still responsible



Headlines and social feeds love a clean narrative: the tool arrives and the human steps offstage. Real life is messier. **No matter how advanced AI gets, there will always be a need for a human to steer it, govern it, and answer for what happens next.** Models can draft, summarize, suggest, and accelerate. They do not carry the weight of choosing what *should* happen when values conflict, data is thin, or a livelihood is on the line. I treat AI like serious infrastructure: powerful, dangerous if mishandled, empty without someone who understands the goal, the constraints, and how much damage a wrong call can do. I build software for a living; I am not allergic to automation. I am allergic to the fantasy that people are optional. The next decade belongs to teams that pair **judgment with leverage**, not the ones that confuse speed with wisdom.

Part five: A friend, Bitcoin, and the sound of a crowded room



Years ago, when money still felt like banks and brokers to most people I knew, a **friend of mine** bought **Bitcoin** early. Back then, explaining it at a party was a good way to watch polite people change the subject. It was not a manifesto about the future of money. It was closer to tinkering with a strange door in the basement of finance: curiosity, a little cash you could afford to lose, and a willingness to look ridiculous if the whole thing faded into nothing. What stayed with me was never a particular candle on a chart. It was how **time changed shape** the moment the buy button was real. A decision that took minutes sat inside years of aftermath: mockery when the crowd thought it was silly, euphoria when the crowd decided it was genius, fear when the same crowd panicked, and headlines that rotted faster than milk. I watched someone learn in public what **volatile** means in the stomach, not on a slide.

If you have never ridden a volatile asset with your own net worth attached, it is hard to describe how quickly the version of events in your head rewrites itself. When price climbs, your brain serves you a clever narrative about vision and patience. When price falls, the same brain serves you a disaster movie, often in the same week. My friend's **investment journey** was not a straight line from wisdom to wealth. It was a loop: conviction, doubt, boredom, excitement, envy, relief, shame, and the slow stretches in between when nothing happened at all and that nothing still felt loud. Mostly as a witness, and later in my own smaller experiments, I saw that **markets are a mirror**. They show you what you believe about risk, about enough, and about whether you can sit still when everyone around you is yelling.

Trading, in the practical sense, showed me what no blog post at two in the morning could. A plan you only remember when you are winning is not a plan. Liquidity is not the same as safety; being able to sell in a second does not mean you will sell wisely. Fees, taxes, and slippage are the dull details that turn a clever trade into math homework. Most people, myself included on a bad day, overestimate how calm they will be when the number on the screen is real money that was supposed to be tomorrow's options.

The hardest lesson was uglier, because it was not about charts at all. **Greed gets the best of us**, and it does not always arrive wearing a villain mask. Sometimes it shows up as “just a little more,” or “everyone else is still in,” or “I will sell after the next leg up.” Sometimes it shows up as refusing to take a win because the win does not feel big enough compared to someone else's highlight reel online. Sometimes it shows up as revenge trading after a loss, which is ego doing risk management with a blindfold on. I watched greed turn smart people into short-term thinkers, not because they were stupid, but because the human brain is built for scarcity alarms and social comparison, and markets feed both on purpose. The antidote is not pretending you are above it. The antidote is rules you write when you are calm, position sizes that let you sleep, and the humility to admit that if the trade needs you to be a saint, you will eventually fail the trade.

I tell this because building and running a business works the same way in your head more than most founders want to admit. You decide with incomplete information. You commit before the crowd agrees. You live with uncertainty while strangers shout certainty at you from timelines and comment sections. Fighting for attention online is as loud as any price chart, and **discipline** is what keeps you from chasing every swing in strategy, every shiny tool, every panic hire, every client who promises exposure instead of payment. Whether your stake is code, reputation, or cash, the same questions apply: know why you are in, know what would change your mind, and do not let strangers live in your head for free.

I am not your financial advisor, and this chapter is not portfolio advice. It is an account of watching a friend live inside a new kind of volatility, and of what I took from it about greed, trading, and the long, uneven walk we call an investment journey.

Part six: Two hats, one head



For a long stretch now I have worn two kinds of responsibility at once. Inside **KAAI TECH LLC** I am allowed to be blunt, move fast, and live with the consequences in a direct line from my keyboard to the customer. Outside it, I also sit in **CTO**-style leadership, including work aligned with **BuyItaly Mortgages**. That second world is not a darker version of the first. It is a different language.

A mortgage is not a landing page where you can A/B test your way out of discomfort. People bring fear, hope, family math, and years of patience into the room. The systems underneath are older than your favorite framework, and the cost of a careless deploy is not only technical. It is trust in a decision that touches someone's life. So I slow down where slowing down is respect, not cowardice. I translate **risk** into choices a non-engineer can actually pick between. I treat **legacy** as debt someone else paid for with real weekends, not as an insult to my taste. And I stop pretending **politics** are beneath the work. In a large organization, alignment is not optional fluff. It is how money and permission move. If you fight that reality with pure idealism, you become the person who is technically right and practically alone.

The skill is context switching without becoming two different humans. At **KAAI TECH** I might push a fix before lunch because if it breaks, I answer for it. In an enterprise lane I might spend the same afternoon helping someone choose between two imperfect paths because the third path is "rewrite the universe" and nobody has budget for a new universe. If you can hold both without sneering at either,

you become useful in rooms where most people pick a tribe and dig in. If you pretend the jobs are identical, you will frustrate executives who need translation, engineers who need clarity, and yourself most of all, because you will keep applying the wrong speed to the wrong problem.

When engineers ask for clarity, they often mean precision down to the semicolon. When leadership asks for clarity, they often mean **priced options** and someone who will stand next to them if the bet goes wrong. I walk in with three sizes: small, medium, and “only if the building is on fire.” Names change. Sometimes I call them plan A, plan B, and plan C. The idea is the same. People relax when they can choose instead of guess, and they trust you more when you attach rough costs and risks in plain language, not when you perform intelligence by drowning them in detail.

Meetings were never my religion, but I stopped treating them as pure waste the day I saw what a **good** one prevents. The best ones I have been part of ended with three boring miracles: a **decision**, an **owner**, and a **date**. Anything else is often a slow transfer of anxiety dressed as collaboration. I still watch meetings that could have been emails, and emails that should have been meetings. The skill is not purity. The skill is ending the ritual with momentum instead of a vague “circle back” that means nobody has to commit yet. **Attention is money**. When you lead technical people, guarding their focus is one of the few leadership acts they will actually thank you for later.

Part seven: Small truths (lightly seasoned)



I keep a few jokes in my pocket because this industry takes itself seriously enough to bruise. They are not there to dodge the work. They are there to keep the ego from mixing up loud opinions with facts.

I once watched a team treat a deployment like a séance. Everyone knew the steps, as long as one specific engineer was in the room, awake, and in a good mood. If your **checklist** lives entirely in one head, you do not have a checklist. You have a **cult** with good intentions. The fix is boring: write it down, test it when that person is on vacation, and accept that redundancy is a feature.

“We will **document it later**” is a sentence I have signed in my mind more times than I want to admit. Later is a mythical country where everyone has free time and no new fires. The polite version is a lie we tell so we can keep moving. The harder part is that later rarely arrives, and when it does, nobody remembers the context. Document the weird part while the scar is fresh. Future you is still you, and they will be tired too.

People love to say the **cloud** absolves you. It does not. It is someone else’s computer, running someone else’s SLA, sending you invoices while **you** still own the outage in the customer’s mind. The cloud can save you from racking servers at two in the morning. It cannot save you from bad architecture, bad secrets hygiene, or bad communication when something breaks anyway.

And yes, sometimes a **meeting** arrives that could have been an email. It will still happen. My rule shifted from moral outrage to outcome: if I am stuck in the room, I leave with a **decision**, not a sequel. Sequels are how organizations simulate progress without spending the currency of commitment.

Humor stays small on purpose. The work is serious. The ego does not have to be.

Part eight: What actually moved the needle

Customers do not fall in love with your internal diagram. They remember whether the thing **worked** when they needed it, whether you answered when something broke, and whether you sounded like a human or a ticket system. I have been proud of elegant architecture that nobody felt. I have been prouder of ugly code that made someone's Tuesday survivable. The diagram only becomes memorable when it maps to pain, usually because it **caused** downtime and now everyone has opinions about boxes and arrows.

What saved me more than cleverness was **hygiene** with no applause attached. Backups that you actually restore on purpose, not only when fate forces you. Secrets that never live in Git, even once, because "once" is how leaks become careers. Dependencies on a rhythm so you are not always surprised by a transitive package drama. Logs you can read when your eyes hurt. None of that gets likes. It keeps you off the worst timelines, the ones where strangers screenshot your mistake and your team stops sleeping. Boring wins. Boring is how you stay in business long enough to tell stories.

Saying no became a product feature in my own head. Every **yes** spreads attention thinner. Early on I treated yes as generosity. Later I saw yes as a loan I took from the people who already trusted me: existing customers, existing products, my own health. A no can sound rude in the moment. Often it is the only direct way to protect quality for the work you already promised.

Reputation does not arrive as a brand deck. It stacks up in order: you **deliver**, you **support**, you **fix**, you repeat, until the market shrugs and says, "that is Kory's shop." That sentence is the whole brand. Everything else is decoration.

Part nine: Vicenza, distance, and showing up anyway



I am based in **Vicenza, Italy**. That shows up in ordinary ways in my week: morning coffee while someone else's workday has not started yet, messages that read colder or hotter than the sender meant because we are not in the same room, and the steady need to explain context even when everyone speaks English, because speaking the same language is not the same as sharing the same assumptions.

Async work rewards a kind of writing I did not value enough early on. I stopped treating silence as agreement. If a detail felt odd, I asked a short question. If a decision mattered, I repeated it back in writing. Those habits felt pedantic until I watched them **save weeks** of rework that came from assumptions nobody voiced because everyone thought someone else already knew.

Building for a **global** audience from one place on the map teaches a specific humility. Your default currency, default privacy instinct, default joke, and default sense of what "reasonable support hours" means are local. They are not universal. A user in another country is not being difficult when their expectations do not match yours. They are being human in their environment. I try to build and release with that in mind: clearer copy, fewer implied defaults, more respect for the fact that trust works differently across borders.

Living far from many clients also made clear that being present for people does not require flying across the ocean every month. It means answering the ticket, sending the status update when you said you would, and saying you missed a deadline plainly, without a pile of excuses. You cannot be everywhere. You can still be steady where you are.

Part ten: Playbooks you can steal



When I am tired, I default to shortcuts. These questions are the guardrails I return to so the shortcut does not become a debt.

Before a build, I ask who the **one user** is this week, not who the imaginary crowd is. Crowds are how scope inflates. One real person keeps the work honest. I ask what **pain** we remove, because features that do not remove pain become features you maintain forever for no reward. I ask what would **embarrass** us if it shipped broken, because embarrassment is an early warning system for the parts we are pretending we already tested.

Before a deploy, I want a **rollback** plan that is not vibes. If we cannot undo it, we should know that before we are heroic. I want a human watching **logs** who is not also doing three other crises at once, because split attention is how misses become incidents. I want a **customer-facing sentence** ready if something slips, because the first message is often what people remember more than the root cause.

Before a yes, I ask what **dies** if we say yes: which project slows, which promise thins, which person pays in evenings. Then I ask whether the cost is **one-time** or **forever**, because forever costs are how small companies accidentally become a pile of commitments nobody can carry.

These are not laws. They are the questions that keep me from lying to myself when I am in a hurry.

Part eleven: If you are starting where I started

If I could hand my past self a folded note, it would not contain motivation. It would contain smaller, harder lessons I paid for in cash and time.

I would tell him to **start smaller than his pride wants**, because pride builds cathedrals on week one and then wonders why the foundation cracked. A small slice that ships teaches you the truth faster than a big plan that lives only in your head.

I would tell him to **charge sooner than his fear wants**, because fear whispers that you need one more feature before you are legitimate. Legitimacy often arrives when someone pays, not when you feel ready.

I would tell him to **write things down while memory is fresh**, especially decisions that felt obvious in the room, and to **put commercial work in a contract before the hours disappear**, because enthusiasm is not a payment method.

I would tell him to keep **one place** where production truth lives, even if it is ugly, because ugly truth beats pretty confusion when something breaks at the wrong hour.

I would tell him to **sleep** before refactoring what he does not understand yet, because fatigue and cleverness combine into a special kind of damage that Git cannot fully undo.

And I would tell him to **laugh at himself at least once a week**, because the job will do it for him eventually. Better to lead the joke than to be the joke without noticing.

Closing

KAAI TECH LLC is not one product. It is a bet that a small, serious team can deliver many useful things if we stay straight about constraints and stubborn about quality.

If you read this far, you are probably building something too. Build it anyway, on purpose, with fewer illusions and better backups than I had on day one.

Thank you for your time.

Kory Kaai

CEO and founder, **KAAI TECH LLC**

<https://korykaai.com> · <https://kaaitech.com>

This book reflects personal experience and opinion. It is not legal, tax, immigration, mortgage, or investment advice. For decisions that affect your business or finances, talk to qualified professionals in your jurisdiction.