

Why Health Systems Should Build an In-House AI Coding Academy That Teaches Clinicians to Ship Software and Spin Up Companies, Then Keeps a Slice of the Royalties and Equity Their Doctors Generate

JUN 28, 2026



1



1

Share



Part I Podcast free on Spotify.



Thoughts on Healthcare Markets & Technology



Part I: Why Health Systems Should Build an In-House AI Coding Academy That Teaches Clinicians to Ship Software & Spin Up Companies, Then Keeps a Slice of the Royalties & Equity Their Doctors Generate

Hospitals pay vendors 6-7 figures yearly for software a frustrated clinician could have built over a weekend. The clinician usually spec'd it out anyway...

▶ Listen now

3 hours ago · Thoughts on Healthcare



Part II Podcast episode for paid subscribers only. Also available on Spotify.

To listen to paid episodes in Apple or Spotify, link your Substack subscription via the show settings on those platforms (instructions inside the Substack app under Subscriptions → Podcast).



Thoughts on Healthcare Markets & Technology

Part II: Why Health Systems Should Build an In-House AI Coding Academy That Teaches Clinicians to Ship Software & Spin Up Companies, Then Keeps a Slice of the Royalties & Equity Their Doctors Generate

Hospitals pay vendors 6-7 figures year for software that their own clinicians could have built over a week and spec'd it out anyway...

▶ Listen now

3 hours ago · Thoughts on Healthcare



Discover more from Thoughts on Healthcare Markets & Technology

Expert analysis of healthcare and life sciences markets, technology, investment, entrepreneurship policy, and AI — for investors, entrepreneurs,...

Over 4,000 subscribers

Enter your email...

Subscribe

By subscribing, you agree Substack's [Terms of Use](#), and acknowledge its [Information Collection Notice](#) and [Privacy Policy](#).

Already have an account? [Sign in](#)

Table of Contents

1. Why a hospital is already sitting on a software goldmine
2. What clinicians actually build the second best software
3. Borrowing the academy model from tumor boards to build a new culture
4. Who owns the thing the cardiologist built
5. The money, meaning royalties, equity, and how to collect
6. The objections that will get thrown across the conference table
7. A rollout that doesn't require a moonshot budget or a ribbon cutting
8. What the whole thing looks like in five years if it actually works

Thoughts on Healthcare Markets & Technology is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

Abstract

Health systems keep paying outside vendors six and seven figures a year for software that their own clinicians could have built, and often built better, because the clinician is the person who actually feels the problem at two in the morning. Agentic coding tools have collapsed the distance between knowing a workflow cold and shipping working code, which means the binding constraint is no longer engineering talent. It

is structure, incentives, and ownership. The argument here is that a provider organization should stand up an internal coding academy modeled on the case-based, peer-reviewed rhythm of academic medicine, think tumor boards, morbidity and mortality conferences, grand rounds, rather than a Silicon Valley bootcamp. Teach clinicians to build with these tools, then strike a pre-agreed deal on royalties and equity so the inside path beats quitting to launch a company on the outside, where the system gets nothing. The precedent already exists in university tech transfer, where the Bayh-Dole Act turned faculty inventions into a multibillion-dollar licensing economy. The same logic applies at the bedside, only faster.

Why a hospital is already sitting on a software company it hasn't noticed

Walk the floor of any large delivery network and start tallying the software it rents. There is the EHR, sure, but that is just the mothership. Bolted onto it are dozens of point solutions, a prior auth tool here, a patient messaging layer there, a sepsis predictor, a bed management dashboard, a nurse scheduling app, a discharge planning thing nobody likes, a virtual sitter platform, a coding assistant for the coders. Each one arrived through procurement, got a multiyear contract, and now bills somewhere between low six figures and the price of a decent attending salary every year. A fair number of them do something a sharp clinician could have specced out on a napkin, because a clinician is usually the person who told the vendor what the workflow needed in the first place. The system is, in a real sense, already running a software portfolio. It just happens to own none of it and pays full retail for all of it.

The thing worth sitting with is what the organization actually controls versus what it leases. It controls the clinical data, oceans of it. It controls the distribution, because its own emergency department and its own clinics and its own operating rooms are the first and most captive customer any health software could ask for. It increasingly controls the intellectual property through the assignment clauses buried in employment agreements, even if it has never once exercised that control on purpose. What it lacks is the one connective thing that turns a frustrated physician with a good idea into a shipped, adopted, maybe even sellable product. That connective tissue is a program. Not a building, not a beanbag.

Here is the slightly embarrassing part. Most provider innovation centers, the ones with the nice logo and the open house, are a beanbag chair, a foosball table, and a press release. They host pitch nights. They occasionally take a small check in a startup that came to them. What they almost never do is teach the people who already work there how to build the thing themselves and then keep a piece of it. So the system will

cheerfully wire a vendor four hundred grand a year to license a tool that a motivated hospitalist could have stood up over a long weekend and too much cafeteria coffee, while that same hospitalist gets a polite no when she asks for time and a sandbox to try. The asymmetry is the whole opportunity.

What clinicians actually build the second you hand them an agentic coding tool

For most of the last decade, the gap between a clinician who knew exactly what was broken and a working piece of software that fixed it was an engineering team, a product manager, and a year. That gap is what killed an enormous amount of digital health. The money was certainly there for a while. Venture funding into the sector climbed to something close to thirty billion dollars at its peak earlier this decade before falling back toward the ten billion range in the leaner years that followed, and a depressing share of what got funded was built by people who could code beautifully but had never written a discharge summary, or by clinicians who understood the problem perfectly but had to outsource the build and slowly watch their vision get sanded down into something generic. The rare combination, the clinician who could also ship, has always been worth a fortune precisely because it almost never existed in one head.

Agentic coding tools change the arithmetic. The point is not that a cardiologist becomes a senior engineer overnight. The point is that the cardiologist can now describe a workflow in plain language, get working code back, run it, see what is wrong, and iterate, which is exactly the loop clinicians already run a hundred times a day at the bedside. One controlled study of a popular coding assistant found developers finishing a benchmark task more than half again as fast. The interesting number is not the speed multiplier for people who already code. It is the binary flip for people who never could. A resident who can articulate a problem precisely, which residents are trained to do, can now produce something that runs.

And the things they build are not toys. They tend to be the small, unglamorous, high-friction stuff that no vendor will ever prioritize because the total addressable market is one annoyed department. A better sign-out tool that does not lose half the handoff. A prior authorization drafter that pre-fills the clinical justification from the chart instead of making someone retype it for the ninth time. A registry that tracks the heart failure population the way the actual heart failure team thinks about it, rather than the way a national dashboard assumes everyone does. A scheduling fix. A bedside calculator that bakes in the institution's own protocol. Because of the interoperability rules pushing systems toward open application interfaces, a lot of this can now plug

straight into the record through standardized app frameworks, which means the homegrown tool does not have to live on a sticky note taped to a workstation. It can actually run inside the workflow where the clinician already lives.

Borrowing the academy model from tumor boards instead of importing bootcamp culture

The instinct, when an organization decides to teach its people to build, is to copy whatever the tech industry does. Hire a couple of ex-startup folks, run a hackathon, give everyone hoodies, talk about failing fast. This mostly does not take inside a hospital, because the culture it is trying to graft onto is not a startup. It is academic medicine, which already has one of the most refined peer-learning systems any profession has ever produced, and clinicians respond to it because they were forged in it.

Think about what already exists. Tumor boards, where a hard case gets put in front of a room of specialists who argue about it until a plan emerges. Morbidity and mortality conferences, where something went wrong and the whole point is to dissect it honestly without anyone getting destroyed, so the next patient does better. Grand rounds. Journal club, where people tear apart a paper's methods over bad sandwiches. This is structured, case-based, peer-reviewed, recurring learning, and clinicians already show up for it without being dragged. An academy that teaches building should sit inside that rhythm rather than fight it.

So the curriculum looks less like a coding bootcamp and more like a longitudinal clinical course. Cohorts, not one-off workshops. Cases, meaning real problems pulled from the floor, not abstract exercises. A recurring forum where someone presents the tool they built and the room interrogates it, which is essentially morbidity and mortality for software, same honest energy, hopefully fewer lawyers. Code review reframed as peer review, which clinicians already understand in their bones. Continuing medical education credit attached, so the time spent counts for something the licensing board cares about. Residents and fellows pulled in early, because the people with the most stamina for learning a new skill are the ones who are already drinking from a firehose and have not yet calcified into the belief that the EHR simply cannot be improved. Faculty mentors who have shipped something get protected time and recognition, the same way a great clinical teacher does, because if building is treated as a hobby done on personal time, only the already-privileged few will ever do it, and the program quietly becomes another perk for the people who least need one.

Who owns the thing the cardiologist built on a Tuesday night

Now the awkward question, the one that makes general counsel reach for the antacids. A hospitalist builds a genuinely useful tool on a Tuesday night, partly on her own time, partly using insight she could only have gotten from treating the institution's patients, possibly touching the institution's data. Who owns it. The honest answer at most organizations today is some combination of nobody knows, it depends on which clause you read, and we will find out expensively if it ever gets valuable.

A lot of people assume the lever here is the noncompete. It mostly is not. The federal effort to ban noncompetes broadly got blocked in court before it ever took effect, so the legal landscape stayed roughly where it was, but the more important point is that the noncompete was never the right instrument anyway. The instrument that actually decides ownership is the intellectual property assignment language in the employment agreement, the part almost nobody reads, which frequently says that inventions related to the scope of employment belong to the employer. Most systems have that language. Almost none of them have a deliberate, fair, transparent process for what happens after it triggers. So the default today is one of two bad outcomes. Either the tool sits in legal limbo, useful to no one because nobody will touch ambiguous ownership, or the clinician, sensing exactly this, quietly quits, takes the idea outside, builds the company on the open market, and the system that incubated the whole thing gets precisely nothing except the bill when it later buys the product back as a vendor.

The fix is not novel. Universities solved a structurally identical problem decades ago with the Bayh-Dole Act, the nineteen eighty law that let institutions own and license inventions that came out of their walls, which is the single legal change that created the modern academic tech-transfer economy more or less from scratch. The lesson is not that the institution should grab everything. It is that clear, pre-agreed ownership with a fair split is what unlocks the value, because clarity is what lets capital and effort flow without fear of a fight later. A clinician will build inside the system if the deal is known in advance and feels fair. She will build outside it if the inside path is a black box that might swallow her work. The entire design problem is making the inside path the obviously better one, on purpose, before anyone has anything worth fighting over.

The money, meaning royalties, equity, and the carry a system can realistically collect

Here is where the academic precedent gets specific and a little intoxicating. University tech transfer typically splits the proceeds of a licensed invention in rough thirds, some to the inventor, some to the inventor's department or lab, some to the central institution, with the exact ratios varying by school. The portfolio behaves like venture, which is to say it follows a brutal power law. The overwhelming majority of disclosures earn nothing. A handful earn a little. And once in a long while one of them pays for the entire enterprise several times over. Stanford's recombinant DNA patents, the foundational gene-splicing work, brought in something on the order of a quarter of a billion dollars over their life before they expired. The shares Stanford received for licensing a certain search algorithm to two graduate students were later worth somewhere around three hundred and thirty-some million dollars. Nobody underwrote those outcomes by spreadsheet. They underwrote the program and let the distribution do its thing.

Provider organizations have already proven the same machine works on their side of the fence. One large clinic's innovation arm has spun out well more than eighty companies over its life, taking equity along the way. One academic system's enterprise vehicle has committed north of a billion dollars to building and backing companies and helped stand up businesses that went public, the kind of population-health platform that started as an internal answer to an internal problem. Another major academic system's innovation operation has booked hundreds of millions in cumulative licensing revenue off inventions that walked in the door as faculty disclosures. None of this is theoretical. The model is old. What is new is that the agentic tooling lets the source of disclosures widen from the small set of clinicians who also happen to be inventors or researchers to, in principle, anyone on staff with a sharp problem and a few weeks of training.

The deal terms matter enormously, and the single biggest mistake would be to design them out of greed. If the system takes the lion's share, the program dies, because the talented people simply route around it, and a clause you cannot enforce against someone who has already left is worth nothing. The split has to leave the clinician with enough to make staying genuinely better than leaving, call it something in the neighborhood of forty to fifty percent of the royalty stream, or a real founder-sized equity stake if the thing becomes a company, with vesting that rewards sticking around. The system's edge is not that it can squeeze. Its edge is that it is customer zero. The hospital is the first buyer, the validation site, the reference account, and the data partner, which de-risks an internal tool in a way no outside startup can replicate, because the outside startup spends its first two years and most of its first round just trying to get a single health system to pilot anything. The system gets that for free on

day one. That advantage, packaged into a fair split, is worth more than any clause that tries to claim everything and ends up collecting nothing.

The objections that will get thrown across the conference table

The first objection is distraction. Clinicians are already underwater, the argument goes, so the last thing anyone should do is hand them a side project. It is a real concern and also slightly backward. A large fraction of clinicians report burnout, and a recurring driver is the documentation and administrative grind, the studies that found physicians spending something like two hours on computer and desk work for every hour of actual patient care. A program that lets the people drowning in that work build the thing that drains the pool is not a distraction from the burnout problem. It is one of the few interventions that treats the cause rather than handing out another resilience webinar. The trick is structure and protected time, so that building is a sanctioned activity for a defined cohort, not a midnight hobby that competes with sleep.

The second objection is the heavy one, patient safety and regulation. Some of what clinicians build will be administrative and low-risk, a scheduling fix that cannot hurt anyone. Some of it will edge toward software that influences clinical decisions, which can land squarely in regulated territory as a medical device, with all the validation, quality systems, and oversight that implies. This is not a reason to refuse to start. It is a reason to put a clinical safety and regulatory gate inside the academy from the first cohort, the same way any responsible research operation has an oversight layer, so that the question of whether a given tool needs formal validation or clearance gets asked early and by people who know the answer, not discovered later by a regulator. Treat the safety review as a feature of the curriculum, not a speed batch bolted on after something already shipped.

Then come the rest, in rapid fire. Privacy and the handling of protected health information, which is solved the way every research environment solves it, with de-identified or properly governed data in a controlled sandbox rather than the live record. Liability, which lives in the same governance layer. Fairness, the worry that only the well-connected clinicians get access, which is exactly why the program must be a real cohort with an open application process rather than an invite list for the chief's friends. And the cultural one, the executive who folds his arms and says we are a care delivery organization, not a tech company. The polite response is that he is already running a sprawling, expensive, externally owned software operation. The only question on the table is whether the organization wants to keep renting all of it

forever or start owning some of it. Compliance lawyers reading this have, understandably, already developed a tension headache. The answer to that headache is to put them in the room from the beginning as architects of the deal rather than as the people who get handed a finished mess and asked to bless it.

A rollout that doesn't require a moonshot budget or a ribbon cutting

The failure mode for anything labeled innovation inside a health system is that it gets announced as a moonshot, gets a renovated floor and a launch event, raises expectations to the moon, ships nothing for eighteen months, and gets quietly defunded in the next budget cycle. The way to avoid that fate is to start almost embarrassingly small and let results, not architecture, earn the next dollar.

A first version needs only a few things. One cohort of motivated clinicians, drawn by application rather than appointment, with a mix of attendings and residents so the energy and the judgment balance each other. A couple of mentors who have actually shipped something and get protected time and credit for teaching. A governed sandbox stocked with de-identified data and the agentic tooling, plus a clear, simple safety and privacy gate that every project passes through. And, before anyone writes a line of code, a plain-language term sheet that says exactly who owns what and how the money splits if a tool gets adopted or spun out, because settling the ownership question while everything is still worthless is the single highest-leverage move available and it costs nothing but a few honest conversations and a lawyer's afternoon.

From there, the discipline is to measure the program by the right thing. Not press releases. Not the number of ideas generated, which is a vanity metric that rewards talking over shipping. The metric is tools built that get adopted inside the organization and improve something the organization already cares about, throughput, cost, quality, clinician hours saved, the boring operational numbers that show up in a board deck anyway. Tie the early projects to those existing goals so the academy is not a sidecar but a contributor to the work the system is already trying to do. Let the customer-zero advantage compound, because every tool that gets adopted internally becomes a validated, reference-backed asset that is far easier to spin out or license later. Build the building, if a building is ever warranted, with the money the first cohort's tools save or earn. Reverse that order and the whole thing becomes another beanbag.

What the whole thing looks like in five years if it actually works

Run this forward a handful of years and the picture is not a unicorn. It is something steadier and arguably more valuable. The organization holds a portfolio of internal tools it owns outright, which means a meaningful chunk of the software it used to rent is now built, maintained, and improved by the people who use it, with the licensing line that used to flow out the door now staying in the building. A few of those tools have grown past the walls into actual companies, with the system holding founder-grade equity it negotiated back when the thing was a side project nobody believed in. The royalty and equity line has gone from zero to a real number on the financials, small relative to patient revenue but high margin and growing, the kind of line a private buyer or a rating agency notices.

The flywheel that matters most, though, is the human one. The clinicians who would have burned out and left, or left to start something elsewhere and taken their best ideas with them, have a reason to stay, because staying is where they get to build, get supported, and get paid fairly when it works. Recruiting pitches itself, because the surgeon weighing two offers now has one of them saying she can build the registry she has been daydreaming about for years and keep a stake in it. The culture shifts from learned helplessness about the software everyone hates toward a quiet expectation that if something is broken, someone in the building can fix it, and might get rich doing so. That shift is hard to put on a spreadsheet and is probably the most valuable thing on this entire list.

None of it requires believing that every clinician becomes a founder, because most will not, and most of what gets built will be modest and useful and never leave the floor, which is completely fine. The model only requires believing the same thing every venture investor and every university tech-transfer office already believes, that you cannot pick the winner in advance, so you fund the program, keep the splits fair enough that talented people actually show up, own a clean piece of the upside, and wait for the distribution to do what distributions do. The hospital is already paying for the software, already sitting on the data, already employing the people who know exactly what is broken. The only thing it has been missing is the nerve to teach them to build it, and the spine to write down a fair deal before anyone knows which Tuesday-night project turns out to be the one worth fighting over

Subscribe to Thoughts on Healthcare Markets & Technology

By Thoughts On Healthcare Markets & Technology · Hundreds of paid subscribers

Expert analysis of healthcare and life sciences markets, technology, investment, entrepreneurship, policy, and AI — for investors, entrepreneurs, hospital and insurance executives, and physicians navigating the business of healthcare.

By subscribing, you agree Substack's [Terms of Use](#), and acknowledge its [Information Collection Notice](#) and [Privacy Policy](#).

[← Previous](#)

Discussion about this post

Comments Restacks



Write a comment...

