

Vignette #2 – The Developer

By Andrew Dana Hudson

Space for Notes

Gerome just wanted to make things smoother. Frictionless. He was old enough to remember the days before tech entirely ate the world and so remembered keenly his first encounters with computer software, the feeling of wonder that code, if you spoke its language, would just do what you told it to at the press of a button. Sure as math. A goat that needed milking or a fence that needed building or a sibling that needed scolding—these were not so accommodating. And so Gerome dreamed of bringing the surety and smoothness of software into the messy world.

Decades later, as the lead dev for a health tech company, that dream was still alive, and yet, as dreams tend to become as we grow older, somehow more distant than ever. For one, the large language models that had been shoved into the heart of modern software were not always the reliable operators that pure code had been. The models were math, yes, but also language, and language was slippery, subtle, imprecise—sometimes more like wrangling the goat or cajoling a sibling than making ten times ten equal one hundred. But all the more reason, Gerome thought, to aspire toward the efficiency and smoothness software could provide, particularly in a strained and high stakes arena like healthcare.

Gerome's company took these stakes seriously. They were not slop-jockeys; they were creating tools that would impact real human lives, often when those lives were at their most fragile and fraught. For Gerome, 'AI safety' was not a matter of some sci-fi apocalypse coming true, but of real issues: patient privacy, data protection, explainability and traceability, the huge liabilities that could be incurred in the case of a false positive or false negative. There were byzantine regulations to navigate and thorny bioethical dilemmas that had to be worked out and instantiated as code. There were the patients, each in need of swift comfort, and there were the clinicians and other workers, to whom the products Gerome was building would be both a tool and a kind of manager. LLMs were notorious for being willing to divulge sensitive information when prompted with a cleverly phrased poem or roleplay scenario, and thus, to comply with security standards, the system had to carefully limit what workers could access and do. In so doing, Gerome found that their product played a part in defining the boundaries of professional roles.

To make sure they didn't fumble this odd managerial discretion thrust upon his software, Gerome talked to a lot of workers. It had always been part of his design process to observe and interview the end-users, to try to understand their experiences and needs. This practice was less in vogue than it had once been,

as the great tech titans increasingly believed they could tell the users of the world what they should do and desire. But that wasn't Gerome's way. He still sat down with various stakeholders: administrators, clinicians, patients. "Walk me through your day," he'd say. "Tell me where you encounter friction."

"Friction" was Gerome's ultimate opponent. It was friction, he thought, that left coughing patients sitting for hours in waiting rooms or ER hallways, filling the air with their germs. It was friction that led to missed appointments or unfilled prescriptions. It was friction behind so many cases of healthcare worker burnout.

Gerome knew well enough that software didn't always remove friction. Often quite the opposite. The frictions of rustling papers had been replaced with new frictions of clacking keyboards, clicking mice, thumping on greasy touchscreens. Learning to navigate a new HIS was a headache no matter how painless the devs strove to make it.

And because the world was not static, software could not be static either. It had to keep up with shifts in governmental policy, medical terminology, GUI trends, pricing changes, operating system updates, hardware upgrades and breakdowns, and the movements (and often enshittification) of third-party platforms that must be interfaced with for various infrastructural tasks. Everything was shifting sand, and some of that sand would always get in the gears and create new frictions. Many an innovative, elegant solution would in time become the next decade's legacy bloat. Which was why Gerome's crusade against friction was turning out to be not a swift, decisive battle but an endless forever war spanning a long career.

Still, progress against friction was being made, ground was being gained—he could feel it. Gerome's company had launched a product to help with exam room charting and note taking. The LLMs, imperfect though they might be, had powerful capacities to transcribe conversation and reduce them down to their essential points. This was a huge opportunity to automate a burdensome task that slowed down exams, clogged up time outside of exams, and distracted from the intimacy of the patient-clinician relationship. Yes, the product would occasionally hallucinate, but if used properly those occasions should (*should*) be rarer than the rate at which a tired nurse might tap the wrong button or mishear a patient's name. Surely the potential efficiency gains were worth pursuing, even if the output still had flaws. Gerome always shook his head at the hypocritical standards to which machines were held. If an automated vehicle was overall ten percent less likely to screw up than the average human, he thought that was more than good enough. But nobody seemed to see that big picture when an autocab ran over a cat. That would make headlines, when the equivalent human error would not even provoke a shrug.

This was the kind of shallow thinking Gerome blamed when, some months after the exam notes AI was deployed, user satisfaction surveys showed significant discontent. One small mistake, and a clinician would refuse to trust the

product's output for weeks after, wasting endless time double-checking what were, statistically, perfectly reliable summaries. Nurses preferred to aggressively mark up and annotate the notes, rather than work with the tool to streamline excess verbiage and correct issues. Patients loved getting more of the doctor's attention, the speedier access to care, but the ones dispensing that care were less enthused.

Gerome at a conference when it finally clicked *why*. He was giving a talk about finding and eliminating friction—his usual keynote, updated with examples from his company's latest deployment. During the Q&A, a woman stood up and asked him if he'd ever encountered a friction that he wanted to keep. "Can't think of any," Gerome replied.

"Then I guess you've never yelled at a toddler to slow down before he hurts himself," the questioner said. "Or sat on the phone with a parent for an hour before getting them to admit they've been getting lost. Or just needed a few minutes to collect your thoughts."

"I don't see the connection."

"You see friction as slowing down delivery—an inefficiency in a fundamentally transactive activity. But to me, as a doctor, friction is how I get a grip on things. Taking exam notes takes time, but that time is part of my diagnostic process. It's time I use to look for connections and spot lacunas. It's how I spot whether a case *should* be quick and transactive, or might turn out to be more complex, might have dimensions that aren't just medical but social, personal, political. The human condition is not friction-free. Health is not a friction-free concept."

Gerome was surprised that the doctor's speech was met with scattered but vigorous applause.

