SKILLPULSE

Exclusive Magazine for leaders building Skills First Organizations



The Game That Changed Everything

DEMYSTIFYING THE FORWARD DEPLOYED ENGINEER

WHAT DOES IT ACTUALLY TAKE TO GROOM IN HOUSE FDE CAPABILITY

SEPTEMBER 2026

SECOND EDITION

ABOUT

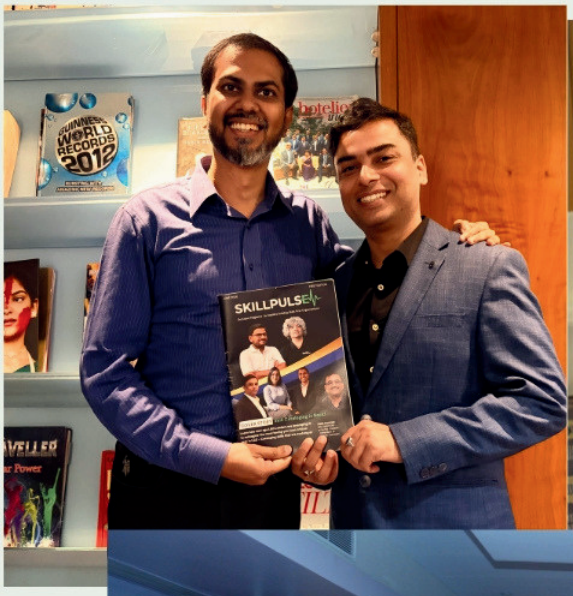
SKILLPULSE

AN INITIATIVE BY ENQURIOUS

SKILLPULSE IS A MAGAZINE AND A COMMUNITY LED BY LEADERS BUILDING SKILLS FIRST ORGANIZATIONS. IT'S A MELTING POT OF IDEAS AND EXPERIENCES ORIGINATING NOT FROM CONSULTANTS BUT ACTUAL PRACTITIONERS SOLVING PROBLEMS ON THE GROUND. EVERY OPINION TRACES BACK TO REAL WORLD CHALLENGES, EVERY RECOMMENDED FRAMEWORK GETS BATTLE TESTED BEFORE IT REACHES THE AUDIENCE. VALUE OVER VANITY STANDS AS THE CORE PRINCIPLE DRIVING THIS COMMUNITY LED INITIATIVE FORWARD.



MEET THE TEAM







SkillPulse Community







Contents

03 **The Engineer Who** *Owns the Outcome*
Vishal Kuchanur
Co-founder and CTO, Varaha

05 **A Trillion-Dollar Opportunity Is Waiting. AI Nativeness Is the Key to Unlocking It**
Amit Choudhary
Founder and CEO, Enqurious

07 THE CIO VIEW
Blue-Collar AI Debate: *Aiming at the Right Layer*
Anupam Singh
Chief Information Officer, Sodexo India

09 **The Young's Modulus of Work**
Arjun Sivasundar
Chief Mentor, Enqurious

11 IN CONVERSATION WITH
The Builders
NileshThakare · AnkitAgarwal · Cladius Fernando · Sudarshan KL
Microsoft Hexaview Hazel AI Quantzig

13 ## The Job We All Used to Do
Jaydeep Chakrabarty
Sr. Director, Piramal Capital & Housing Finance

16 GUEST COLUMN
Demystifying the *Forward Deployed Engineer*
Satwik Muthappa P
AI Academy Leader, KPIT

18 LATEST ON ENQURIOUS
The First Client We Forward Deployed Into Was *Ourselves*
The Enqurious Team

20 ## Subscribe to SkillPulse
enqurious.com/skillpulse

THE LEAD COLUMN

The Engineer Who *Owns the Outcome*

The CTO of Varaha on what he hires for now, why the title arrived this year, and how you put back the repetitions the agents took away.

VISHAL
KUCHANUR

CO-FOUNDER AND CTO
Varaha



I stopped hiring by layer. I used to open roles for backend, data and mobile engineers, slotting people into neat technical buckets. Now I hire someone to own an outcome and know when the AI agents' output is wrong.

That last part is the hard one. Early in a career an engineer knows very little and is humble about it, and much later they've watched enough things break to be careful. In between there's a stretch I call the **valley of competence**, where their opinions are technically correct and they're wrong about the context, and from inside the valley the two look identical. Owning an outcome means being out the other side of that valley. It used to take years to get there.

WHAT AN FDE ACTUALLY IS

The industry has a name this year for the engineer who is out the other side. A Forward Deployed Engineer embeds inside a client's organisation, writes production code against the client's real systems, and is accountable for whether the deployed thing actually works and gets used. That ownership is what separates the role from consulting. It's also why the title has spread so fast: the hardest part of enterprise AI turned out not to be building the model. It's making the model work inside a real organisation.

I treat my engineers as FDEs because I ask them to own the outcome, and the reason that's become the job rather than a line in a job description is that shipping stopped being scarce. AI agents handle most of that layer now. Shipping still matters, but nobody pays a premium for it any more, and once shipping is cheap the only thing left to sell is the outcome, which someone still has to own. You can check whether that's already true for you: look at what your last three statements of work actually charged for. Every engineer just became the tech lead of a team of AI agents, whether or not they applied for the job.

THREE SKILLS, ALL LANDING ON DAY ONE

A good FDE needs three things: technical, domain and people skills. None of them are new. What's new is that all three land on day one. Engineers used to absorb the domain slowly, over years of building the layers of a system. The agents took those years away.

Every engineer just became the *tech lead of a team of AI agents,* whether or not they applied for the job.

The technical skill is now mostly reading and judging. Engineers write less code and read far more of it, and every machine output is a pull request from a fast, overconfident junior. The question moves from whether the code runs to how they'd know if it were wrong. Try this on your own team: hand someone a working design and ask what breaks. Time the pause. The good ones have started answering before you've finished the question, and they write the eval before they touch the feature.

Domain skill means knowing what a wrong answer costs. The model knows the code but has no way of knowing what the code costs when it's wrong. My team builds the software behind carbon projects, so for us an outcome looks like whether a farmer actually got onboarded, or whether the number we sent the auditor held up. Once a number leaves the building with a client's name on it, the cost of being wrong is what matters. The signal I weight most is when an engineer asks what a wrong answer costs before they ask how to get the right one. A client will ask for exactly the wrong thing in perfectly clear language. The machine will build it beautifully by Thursday. **I want the engineer who pushes back on something that's technically correct.**

Then there are the people skills. Every engineer has reports now, and none of them are human. Briefing an agent, checking its work, giving it feedback: that's management, and almost nobody was trained for it. The agents don't sleep and don't argue, which sounds like a dream until you notice they also never tell anyone they're wrong. Owning an outcome means talking to the customer themselves. It means explaining a technical trade-off to someone who doesn't write code, and keeping their name on the system when the agent breaks it at 2 AM.

PUTTING THE REPETITIONS BACK

These skills are built by repetition, by being wrong in front of someone who notices, and the agents took those repetitions away, so you put them back deliberately. Make your engineers ship something end to end with an agent every week, and put them in front of whoever actually uses what they built for a day each quarter, rather than the client's project manager. One incident, owned from the page to the postmortem, sitting next to someone who has cleaned one up before.

> **“ And send people to the client before they're senior enough to be trusted there. *That's exactly when it changes them.* Wait until they've earned it, and you've built a very good engineer who has never once been wrong in front of a client.**
>
> VISHAL KUCHANUR, CO-FOUNDER AND CTO, VARAHA

AMIT CHOUDHARY

FOUNDER AND CEO, ENQURIOUS



“A Trillion-Dollar Opportunity Is Waiting. AI Nativeness Is the Key to Unlocking It

The human race stands at a decisive juncture. For the first time a species that is not natural, created by humans themselves, threatens the very existence of the human race, and at the same time offers a growth opportunity worth trillions of dollars. That is the double-edged sword of artificial intelligence.

McKinsey estimates that across 63 analysed corporate use cases AI could inject up to $4.4 trillion a year into the global economy, and that generative AI could automate activities absorbing 60% to 70% of employees' time today.

That same value threatens to obliterate the employed workforce. Oracle has laid off 30,000 staff. Amazon has cut 20,000 roles, more than a third of them high-end software jobs considered aspirational and safe until now. That is the brutal dichotomy we are all staring at.

The new star kid in town

Coming from a capability development background, I force myself to look at the other side. AI, like every technology before it, casts doom on many job categories and quietly creates new ones.

AI services is the new hot favourite of investors and the Fortune 500 alike, projected to grow from $20 billion in 2025 to almost $100 billion by 2030, a growth rate of 37% to 43% a year.

That cannot be met by a workforce that picks up one more training. It needs a foundational capability change. Which brings me to the new star kid in town: the Forward Deployed Engineer.

- FDE hiring grew more than 1,000% year on year into 2026, with postings jumping roughly 800% in a single year.
- A 2026 FDE Academy study put employer demand at 95% against a talent supply of 45%, the widest gap of any AI deployment role.
- In 2026, 41% of AI engineers spent more than 30% of their time in customer-facing work, up from 12% in 2023.

Why this technology is different

For the first time in history we are dealing with a non-linear, self-healing, self-improving technology. Until now every shift was under human control: every change pre-programmed, every output verified before it was applied.

AI is not the same. It mimics the human way of thinking and working. It thinks for itself, takes action, takes feedback, loops through errors, corrects them and works out the next step until the goal is met, without human intervention.

To business leaders that sounds like music. The cost of AI intelligence has dropped far below the cost of human intelligence, and it buys a system that does not tire, does not argue and does not take leave. That explains the layoffs.

But LLMs, like human brains, are probabilistic: for a given input a model can produce a given outcome, not will. The systems we built until now were deterministic, where the same logic always produced the same output. In supply chain and logistics there is a vast difference between a consignment that must reach its destination on time and one that may.

Here is the bad news and the good news. Nobody has yet worked out how to make an intrinsically

stochastic model produce deterministic outcomes. Bad, because without it enterprises will not adopt AI: read the MIT report on the 95% failure rate. Good, because that is exactly where the opportunity comes from.

Three layers, three different resets

Thisis the pivotal moment forL&D teams and corporate academies. My conversations with leaders across SIs, GCCs and product companies point to a build that works at three distinct levels.

1. **Leadership**. The era of hands-on leadership is back. You cannot throw this technology at your people and tell them to use it. It is a different animal to tame, with a heavy downside if it goes wrong. Leaders are going back to the classroom.
2. **Mid-management**. This layer holds the key to success or failure. It has to learn the technology

furiously, find where it applies in real workflows,



convince leadership that AI will not solve everything, and mentor juniors on using it responsibly.

3. **The ground workforce.** These are the people who must reskill or cross-skill to stay relevant. A Java engineer, a BI engineer or a business analyst has to become an AI engineer, and in no time at all. Building with AI demands a different mindset, skillset and toolset. It needs a reset.

How to build it

Stop buildingcourses and start handing people AI tools to experiment with. This technology cannot be taught with slides. More experiments, more failures. More failures, more learning.

The technology is adaptive, and so should be the people learning it.

Move beyond tech and sharpen domain skills. AI codes and debugs faster than all of us. With the tech layer automated, the difference lies in understanding the domain problems you solve: enterprise workflows across BFSI, retail, healthcare and more.

Groom people in soft skills early. Communication, negotiation, reasoning and storytelling are no longer a one-off training. The engineer who builds a prototype while talking a customer through their real pain points beats the architect who drafts a spec.

AI native skilling shatters silos. It blends tech, domain and soft skills to groom the generation that will finally deliver the value everyone is craving. Happy skilling!

Blue-Collar AI Debate: *Aiming at the Right Layer*

Why AI for India's blue-collar workforce should begin one layer above the floor.

ANUPAM
SINGH

CHIEF INFORMATION OFFICER & DIRECTOR
Technology, Data, Digital and Innovation · Sodexo India



When boards, policymakers and journalists ask how we are bringing artificial intelligence to our blue-collar workforce, they are usually expecting a particular kind of story: a housekeeping attendant with a smartphone in hand. My answer disappoints that expectation.

This is not a retreat from ambition. It is an argument about sequencing. The question worth asking is not how quickly we can place a chat interface in a frontline worker's hand. It is where in the operating chain a single unit of intelligence produces the greatest downstream benefit for the least disruption. In our business, that point sits one layer above the floor.

Read the operating model before designing the intervention

We embed a full team, led by a site manager, inside the client's own premises. In systems terms, that person is the integration layer. They manage the frontline team and reconcile a constant stream of inputs: verbal complaints in three languages, WhatsApp photographs of a leaking pipe, and a folder of spreadsheets that quietly functions as our rostering system.

Most enterprise AI programmes fail not because the models underperform but because the intervention is aimed at a layer of the organisation that has neither the authority nor the bandwidth to convert assistance into outcome.

What the frontline day actually contains

Consider the day of a housekeeping attendant on a 6 a.m. shift. She may not own a smartphone. If she does, it is often a shared household device, left in a locker because the client's site policy requires it.

The barrier is not intelligence, aptitude or willingness. It is the absence of a moment in the working day into which a conversational interface can be inserted. A skilling programme that teaches prompt craft to a person who does not own the endpoint is not inclusion.

It is theatre, and frontline workers recognise it as such faster than anyone in the room.

So we invert the design. If the worker cannot come to the tool, the benefit must be routed to the worker through channels that already exist in their day: the supervisor's briefing, the instruction sheet, the voice note on a basic handset.

Three loops: assisted, ambient, autonomous

- **The assisted loop** is where we are operating today. The site manager uses a sanctioned AI assistant directly. The human remains the author; the model removes the blank page.
- **The ambient loop** is where the frontline benefit actually lands. AI never appears as a tool the worker must operate. It appears as a better instruction, a fairer roster, a clearer briefing, a voice message in the language they actually speak. The intelligence is upstream; the experience is ordinary.
- **The autonomous loop**, agents executing multi-step operational workflows, is a 2027 conversation for us.

The unglamorous prerequisite

None of this is limited by model capability. It is limited by the state of our own data. AI readiness, at this layer of the economy, is very largely data readiness wearing a better suit.

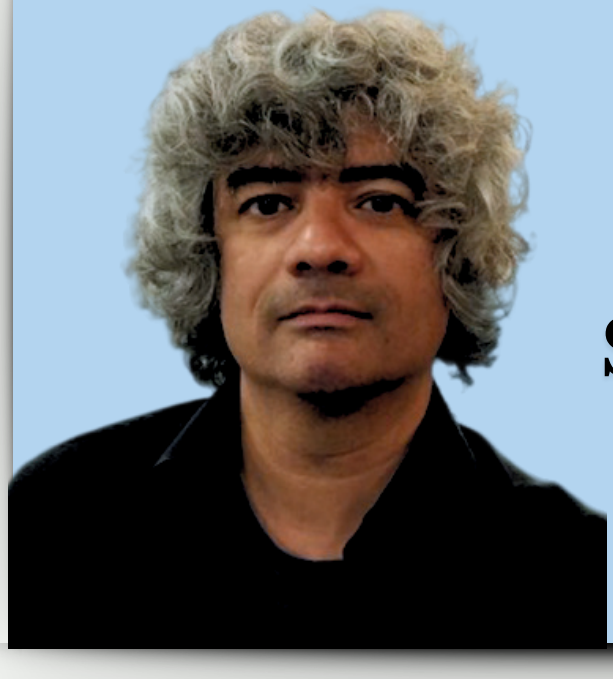
The line I will not cross

The same architecture that helps a manager draft a roster can, with a small shift in intent, become an instrument for continuous surveillance of the workforce. The technical distance between augmentation and instrumentation is short. The ethical distance is not.

The test that counts

The attendant on the 6 a.m. shift may never open an AI tool. If we do this correctly, her day changes anyway. The instruction on the wall is in a language she reads. A voice note tells her that her shift has moved, in her own language, before she makes the journey. She will not know that a model was involved, and she does not need to. ■

> “ Our position is that we *measure work, not workers.*



ARJUN
SIVASUNDAR

CHIEF MENTOR, ENQURIOUS

The Young's Modulus of Work: What a 200-Year-Old Physics Formula Accidentally Explains About Your Job

There's a number engineers use to describe how stiff a material is. It's called Young's Modulus, and the formula is almost insultingly simple:

E = stress / strain

Strain is how much a material deforms under load - how far it stretches, bends, or compresses. Stress is the internal force per unit area that builds up because of that deformation. A low modulus means a material can strain a lot without much stress building up inside it - think rubber. A high modulus means even a small strain generates enormous internal stress - think glass. Stretch rubber and it stretches. Stretch glass and it shatters.

In mechanics, external force (stress) causes deformation (strain). In the workplace, we often experience the inverse: putting in real effort (strain) causes a massive rise in internal friction (stress). Consider the physics inverted on purpose, not by accident. I have interpreted strain and stress as workplace forces, not mechanical.

I think most of us are working in high-modulus jobs. And I think that number - let's call it the Young's Modulus of Work - is quietly the best measure we have of what's actually burning people out.

Strain is not the problem

Strain, in this analogy, is the real work. The hours spent doing the thing you're actually good at. Writing the code, closing the deal, designing the building, treating the patient. It's physically or mentally demanding, sometimes exhausting, but it's load-bearing exhaustion. It has a point. You can point to what it produced.

Most people can absorb a remarkable amount of strain. Long hours in service of real output are tiring, but they're rarely what breaks someone. Athletes train for hours a day. Surgeons stand for hours in an OR. Founders work absurd weeks. Strain alone doesn't explain the specific, hollowed-out quality of modern burnout.

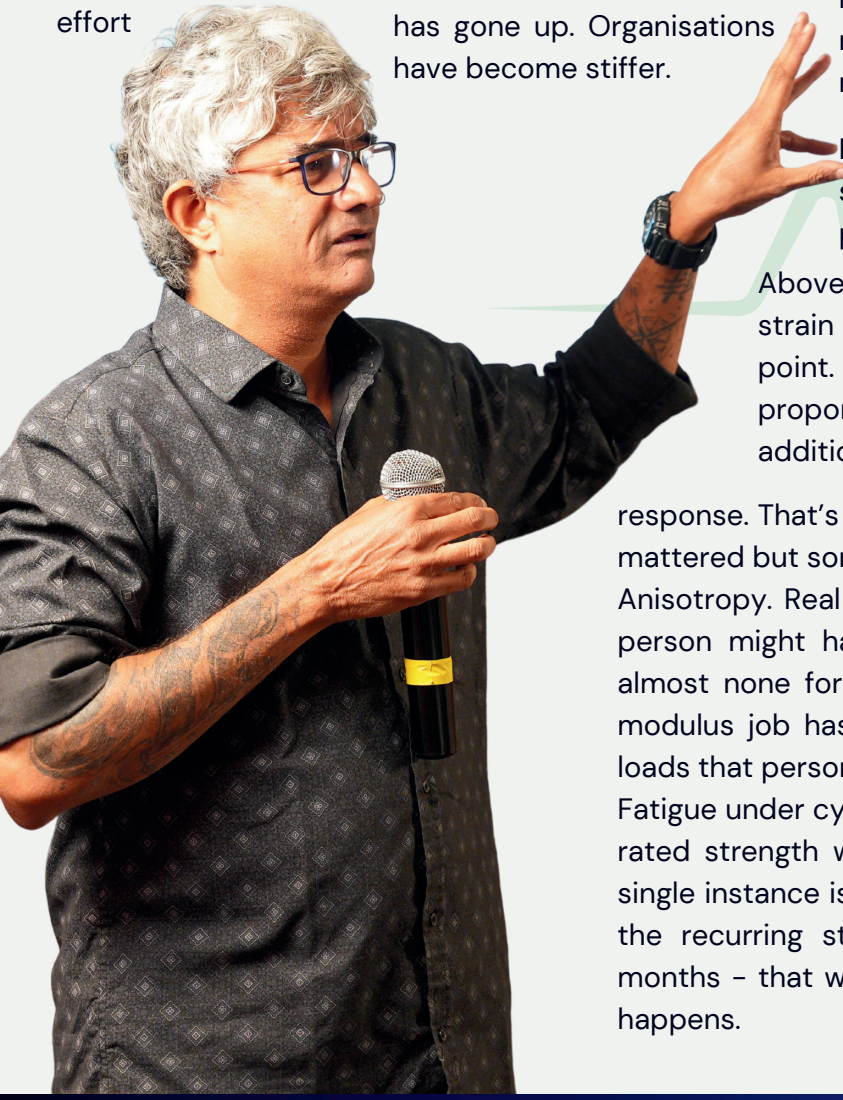
Stress is the problem

Stress is different. It's not the work. It's all the annoying friction around the work. The meeting that could have been an email. The re-org that changes your reporting line for the third time this year. The tool that doesn't talk to the other tool. The performance review process that takes longer than the performance. The pain of getting a straightforward decision approved by six stakeholders who have six different agendas.

None of this is what anyone was hired to do. It's the internal force generated by the system in response to normal effort - and it's the part that doesn't show up on any output report, because it doesn't produce anything. It just accumulates.

Today, stress has outpaced strain

That's the diagnosis. Somewhere along the way, the ratio flipped. It's not that people are working physically or mentally harder at their craft thanthey used to - plenty of periods in history had brutal hours. It's that the amount of internal, non-load-bearing stress generated per unit of real effort has gone up. Organisations have become stiffer.



Small strains - a delayed reply, a missed sync, a change in priorities - now generate outsized internal stress, because there's not enough give in the system to absorb them.

That's a high Young's Modulus of Work. And high-modulus materials share an unglamorous trait: they don't bend. They hold their shape right up until the point they don't, and then they fail suddenly, not gradually.

The goal isn't less strain. It's a lower modulus

This is the part worth sitting with, because it reframes the usual advice. The instinctive fix for burnout is "do less" - work fewer hours, take fewer meetings, lower the strain. Sometimes that's right. But it treats the symptom, not the material property.

The better goal is a low modulus: a role, a team, or an organization where you can put in real, honest effort and it doesn't generate disproportionate internal stress. Not zero strain - plenty of strain, even - but elastic, absorbed, recoverable strain, rather than stress that compounds and never fully releases. A few ideas from materials science that map onto this almost too well:

Every material has a point past which deformation stops being elastic (recoverable) and becomes plastic (permanent). Below it, you bounce back. Above it, you're bent out of shape for good. Burnout isn't strain - it's stress that pushed someone past their yield point. Hooke's Law breaks down. Stress and strain are proportional only up to a limit. Past that, a small additional load produces a wildly disproportionate response. That's the "one more Slack message" that shouldn't have mattered but somehow was the one that broke the day.

Anisotropy. Real materials aren't equally stiff in every direction. A person might have huge tolerance for hard creative strain and almost none for bureaucratic nonsense - or the reverse. A low-modulus job has to be low-modulus along the axis that actually loads that person, not just on average.

Fatigue under cyclic loading. Materials fail at lower stress than their rated strength when the load repeats over and over, even if no single instance is severe. This is the chronic, low-grade nonsense - the recurring status meeting, the same argument every three months - that wears people down well before anything "big" ever happens.

IN CONVERSATION WITH

THE BUILDERS

The people who build capability for a living, in conversation with **Amit Choudhary**

Since June, the podcast has been sitting down with them one at a time. A customer success leader at Microsoft. A founder who has bootstrapped his company for sixteen years. An architect who left R&D to go and teach. And, next up, a learning designer.

They come at it from very different rooms: a global software company, a 450-engineer services firm, an academy serving a 29,000-person organisation. Yet three conversations in, **they keep landing on the same point**, and it has very little to do with the model.



Customer success leader, Microsoft

The series opened in the Enqurious studio, with the first edition of this magazine propped on the desk between the two chairs. Nilesh brings more than twenty years in customer success and technology to the conversation, and he and Amit spent it on one question: what it really takes to move AI from experimentation to measurable business impact.

> ***Successful AI adoption is less about technology alone and more about aligning initiatives with clear business outcomes, driving execution, and building the right skills.***



Founder and CTO, Hexaview Technologies

Ankit started Hexaview in 2010 out of a small room with one partner, and that partner brought him his first customer: Bank of New York Mellon. Sixteen years on it is 450 engineers, still bootstrapped, still profitable. His case on the show is that enterprise AI stalls on legacy, not on the model. His AI University builds skillset and toolset; mindset, he says, has to come with the person.

> ***Enterprises have spent millions and millions of dollars on systems that they have been using for the last 40 years, and they can't just throw it away.***

THE COMMON THREAD

The model was never *the hard part.*

Each guest named a different bottleneck, and each one sits well away from the model.

CLADIUS FERNANDO



Head of AI Academy, CTO, Hazel AI

Cladius joined Persistent in 2005, when it was around 2,500 people. It is 29,000 strong now. He spent years in the CTO's team building multi-agent workflows before anyone called them agentic, then took on the AI Academy. He explains the move through Top Gun: Maverick could have gone anywhere, and chose to go back and train the next batch.

> ***Instead of having a one size fits all course for everybody, what I'm trying to do is meet people where they are.***

SUDARSHAN KL



President, Quantzig

Sudarshan shared his screen for most of two hours, failed experiments included. His AI bill had climbed to twice his grocery bill, so he set himself a constraint: cut it to a tenth. That forced him to learn how models actually handle context. He now runs 800 million tokens across four models for under $70.

> ***If you develop a deck in Opus, I can do the same thing. Anyone in the world can do the same thing. So where's your contribution?***

THE SERIES CONTINUES

Conversations with the people building capability for the AI-first workforce.

ENQURIOUS.COM

THE JOB WE ALL USED TO DO

THE SILENCE IN THE HIRING ROOM

The hottest job title in software describes work that was being done before most of its current holders were born. I know this because I recently sat in a hiring discussion where a leadership team agreed, with great enthusiasm, to recruit three Forward Deployed Engineers by the next quarter, and then fell into a thoughtful silence when someone asked what, precisely, one is.

It was not an ignorant silence. It was the silence of twelve intelligent people who had collectively shipped decades of software and could feel, somewhere behind the ribs, that they already knew the answer and were embarrassed to say it out loud. Because the honest answer sounds like an accusation: it is the job we all used to do, before we divided it into departments.

THE FASHIONABLE BELIEF

The prevailing story goes like this. Somewhere between the research lab and the enterprise, a new species of engineer has evolved. This creature parachutes into a customer's building, absorbs the business, wields AI models the way previous generations wielded compilers, and produces working software before the visitor badge expires. The species is rare, the story continues, and therefore expensive; and because it is rare and expensive, it must be new. Conferences have adopted the term. Job boards glow with it. Salary bands have been adjusted with the solemnity normally reserved for central bank announcements. And to be fair to the enthusiasts, the story is not wrong about the value. It is wrong about the novelty, and the difference between those two errors turns out to matter a great deal.

FIVE QUESTIONS, FIVE OLD ANSWERS

There is a useful test for whether you understand something: try to list the questions whose answers would define it. Not answers, questions. It is harder than it sounds, and more revealing. Applied to the Forward Deployed Engineer, the test takes about five minutes and produces a small heresy.



JAYDEEP
CHAKRABARTY

SR. DIRECTOR, PIRAMAL FINANCE

- **Where does this engineer work?** At client's site, inside the client's world, in the room where the problem actually lives rather than three abstraction layers away from it. Field engineers and solution consultants have occupied that room for decades.
- **What problem do they solve?** Not stated requirement but the real problem hiding behind it, the one the requirement was a clumsy attempt to describe. That has been the business analyst's entire reason for existing since the profession was invented.
- **What do they deliver?** What do they deliver? understood, designed, built, shipped, and prepared for growth. That is not a novel artefact. That is the software delivery lifecycle, drawn on a thousand whiteboards.
- **What skills must one person carry?** Discovery, architecture, code, operations, and the ability to explain all four to a sceptical finance director. The same skills, in other words, that we have always needed; we simply distributed them across an entire delivery team and called the arrangement an operating model.
- **When is the job done?** When the client can run the thing without you. Walk in, solve, hand over, walk out. Consultants were being paid for exactly this before the first web server drew breath. Five defining questions. Five answers, every one of them answered long back. So the interesting question is not what a Forward Deployed Engineer is. The interesting question is why a role that always existed suddenly acquired a name, a salary premium, and a fan club.

What actually changed?

The answer is economics, and it is worth being precise about which economics.

A software solution is produced in stages, and the stages are not equally expensive. Understanding the client's world and locating the real problem are what I would call question stages: their cost is mostly judgment, attention, and the willingness to sit in uncomfortable meetings. Design, build, ship, and scale are answer stages: historically their cost was labour, and lots of it. An architect for the design. A team of developers for the build, sprint after sprint. A release process for the shipping, with its ceremonies and its change boards. A capacity-planning cycle for the scaling. No single human could afford to do all of it, so no single job title could hold the work, and the lifecycle was carved into professions the way a large estate is carved into inheritances.

Then the cost structure collapsed. AI did not make the work different; it made the answer stages cheap. A design that once consumed three weeks of architecture review can now be drafted, challenged, and redrafted in an afternoon, with the human doing the challenging rather than the drawing. Code that required a team now requires a conversation and a code review. Deployment pipelines had already automated themselves into the background. Suddenly the entire lifecycle fits inside one person, not because that person is a genius, but because the expensive parts stopped being expensive.



When a job becomes affordable for one person, the market invents a title for that person. That is the whole mystery. The trumpets and the think-pieces are announcing a price change.

Experienced practitioners will notice what the price change did not touch. The question stages cost exactly what they always cost. No model, however fluent, walks into a warehouse and notices that the stated problem about inventory dashboards is actually a problem about a purchasing process nobody trusts. The machine produces magnificent answers; it has no opinion about whether the question deserved one. Which points at the genuinely dangerous failure mode of this era. It is not the model inventing nonsense; teams have learned to check for that. It is the model producing a confident, competent,

entirely correct answer to the wrong question, and the organisation shipping it with pride. A wrong answer gets caught in testing. A right answer to the wrong problem sails through every quality gate you own, because every gate was built to inspect the answer.

Four principles for the price collapse

1

Staff the question stages deliberately

Interview for problem-finding, not syntax. Put your most contextually fluent people at the client boundary, where the questions are formed, rather than promoting them away from it.

The common misunderstanding is to treat discovery as a junior activity that precedes the real work; it is the real work, and everything after it is increasingly a commodity.

2

Reprice the answer stages honestly

Stop staffing 2015-sized delivery teams for work the tooling has deflated, and stop letting estimates inherit their ancestors' assumptions.

The misunderstanding to avoid: cheap drafts are not free decisions. A generated architecture still needs a human who can say what breaks at ten times the load, because the model will not volunteer the question.

3

Keep the customer in the room

If the long arc of this role ends with machines doing the stages, then the person stating the problem becomes the engineer of record, and that person is the customer. Treat every engagement as an apprenticeship in problem statement: outcomes rather than features, constraints spoken aloud, success defined before work begins.

The misunderstanding is to see this as customer education, a soft activity for the account team. It is succession planning.

4

Define done as departure

A forward deployed engagement that cannot end has quietly become an outsourcing contract with better branding. Handover is the deliverable; the solution is merely its container.

Misunderstood, this principle turns into abandonment. The point is not to leave quickly but to build the client's ability to be left.

GUEST COLUMN

Demystifying the *Forward Deployed Engineer*

What it actually takes to groom in-house FDE capability.

BY SATWIK MUTHAPPA P

SATWIK
MUTHAPPA P

AI ACADEMY LEADER
KPIT



There is a moment I keep returning to. A young engineer on my team had just spent few weeks building a beautiful Agentic AI solution. Elegant architecture. Clean prompts. An impressive demo. And when we took it to the business team it was built for, one of them looked up and said, quietly, "This is lovely. But this isn't how we actually work."

I watched his face fall. And then I watched something better happen. He pulled up a chair, closed his laptop, and said, "Can you show me how you actually work?"

He spent the next couple of days sitting with that member. Not building. Just watching, asking, listening. When he came back and rebuilt the solution, it was less elegant and infinitely more useful. Adoption went from zero to near-total in a fortnight.

That was the day I stopped thinking of Forward Deployed Engineers as a role. **They are a posture.**

The myth we need to retire

The industry has romanticized the FDE into something almost mythical: a genius engineer parachuted into a client site, solving impossible problems by sheer technical brilliance. That story is seductive. It is also mostly wrong.

What I have learned building this capability in-house is that the hardest part of the FDE is not the "engineer." It is the "forward deployed." It is the willingness to leave the comfort of your own function, sit in someone else's chaos, and stay there long enough to earn the right to solve something.

Most organizations try to hire this. My conviction is that you have to grow it because the raw material you need is already sitting inside your company, in people who understand your workflows, your language, your unwritten rules, and your customers. **You are not recruiting FDEs. You are releasing them.**

You are not recruiting FDEs. You are releasing them.

- **Find the curious, not just the capable.**
  The best FDE candidates I have seen were rarely the most technically decorated. They were the ones who asked one more question than everyone else who were genuinely, almost annoyingly, interested in how other people's work happened. Technical depth can be taught. Curiosity is a temperament. Look for it, then protect it fiercely.
- **Teach them to sit in on the problem before reaching for the solution.**
  Engineers are trained to build. FDEs must be trained to wait to spend the first week not building a single component, mapping the actual workflow, including the messy workarounds people are slightly embarrassed to admit they use. Every FDE I have grown has had to unlearn the instinct to rescue before they understood.
- **Let them fail where it is safe, and stay where it is hard.**
  Rotations of two weeks produce tourists. Real FDE capability is built in three-to-six month embeddings, where they live with the consequences of their own design choices. That is where the learning compounds.
- **Give them business fluency, not business briefings.**
  An FDE must be able to sit with a business head and speak in outcomes, cycle time, quality, cost, and customer experience not the tokens to be consumed, models and architectures. Rotate them through business reviews. Let them own an outcome metric. Fluency comes from immersion, never from a deck.
- **Build trust muscle deliberately.**
  This is the part nobody puts in the job description. An FDE walks into rooms where people are quietly anxious that AI has come to replace them. Their first deliverable is not a solution. It is safety. Coach for this explicitly how to listen without defending, how to celebrate the team's expertise, how to make people co-authors rather than recipients.
- **Bring them home to teach.**
  The single highest-return practice we have adopted: every FDE returns to the academy and teaches what they learned. It multiplies capability in-house, and it tells the organization that this path is valued.

Why this matters more than we admit

Here is what moves me about this work. We talk endlessly about AI adoption as a technology challenge. It is not. It is a human one. It fails in the quiet gap between a brilliant solution and a person who does not yet believe it is for them.

The Forward Deployed Engineer exists to close that gap with patience, humility and proximity. They are, in the truest sense, translators between what technology can do and what people are ready to receive.

And when you grow them in-house, something else happens that no external hire can replicate. Careers change. People who never saw themselves as transformation leaders suddenly are. Confidence spreads. The organization starts to believe it can build its own future rather than buy it.

That is not a capability programme. That is a culture shift.

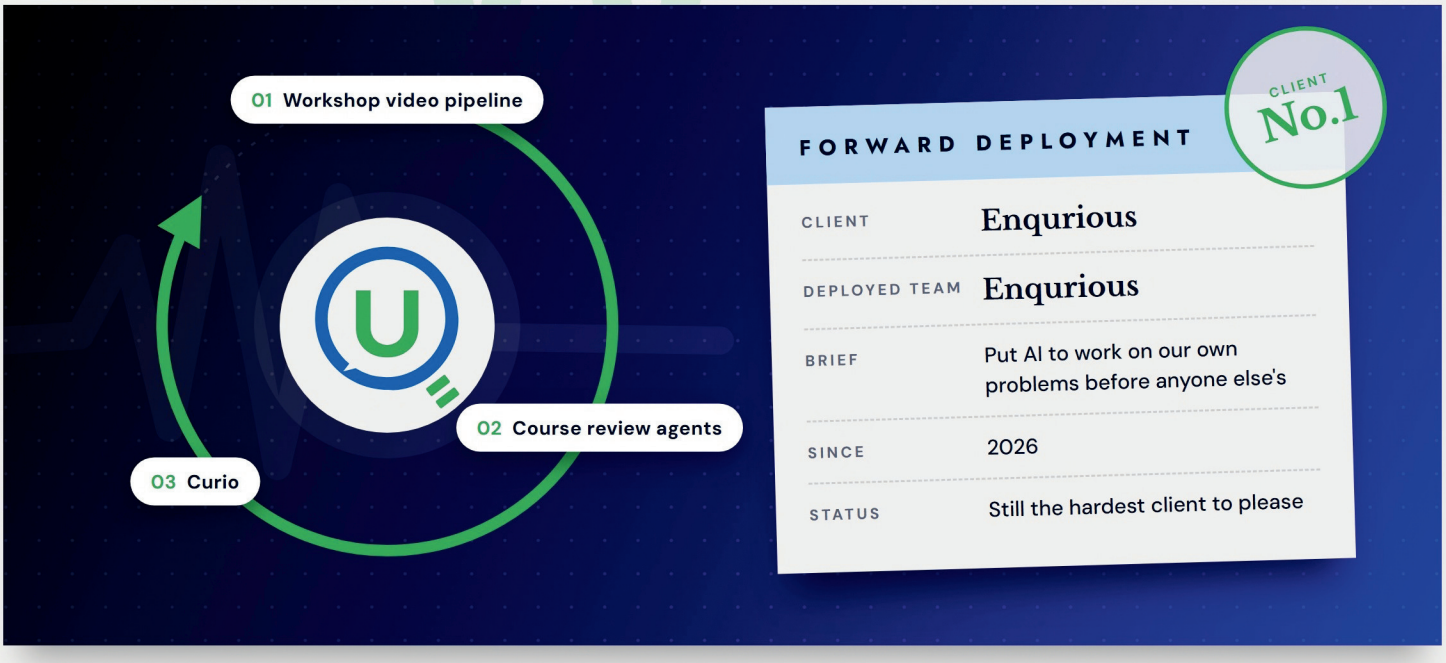
Build the FDE, and you do not just deploy AI. You deploy belief and belief, in my experience, is the only thing that ever truly scales.

LATEST ON ENQURIOUS

The first client we forward deployed into was *ourselves.*

What changed when Enqurious put AI to work on its own problems before taking it to anyone else's.

BY THE ENQURIOUS TEAM



One of us joined Oracle in 2007, in Oracle Solution Services, which was more or less what Databricks Professional Services is today. Go to the client, sit with the problem, build, hand over, leave. So when the industry started calling this the **Forward Deployed Engineer**, our honest reaction was that we have seen this movie before.

What is new is the suit. AI works like an exoskeleton. It lets one person, or a small pod, carry work that used to need an analyst, a data scientist, two developers and a consultant. But an exoskeleton only amplifies whoever is wearing it. We had been saying that to clients for a while. This year we checked whether it was true of us.

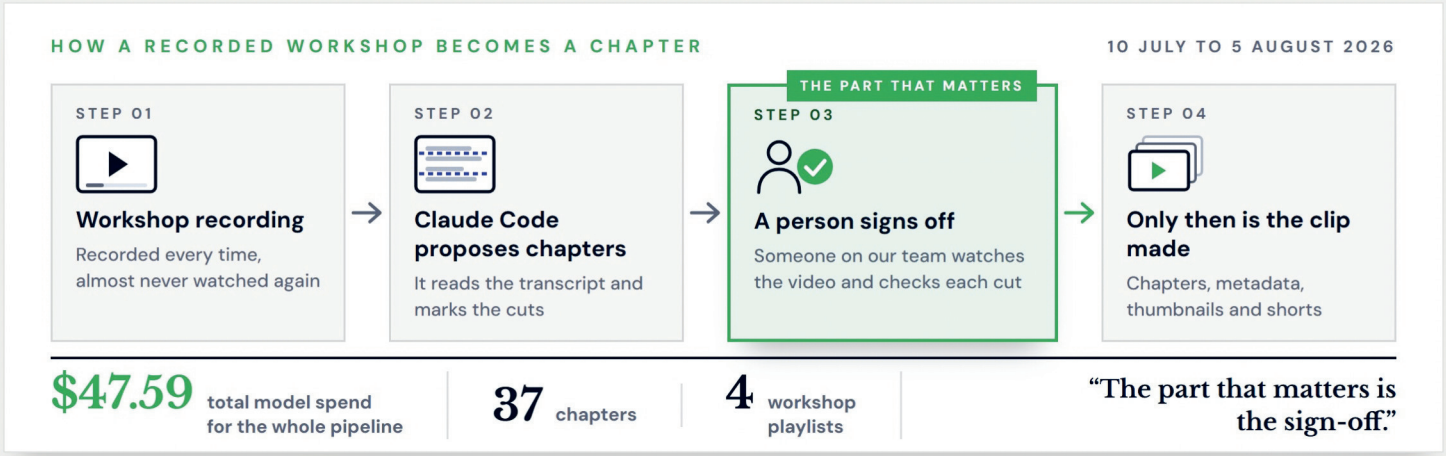
Our own mess came first

Every training company has the same dark data. We run instructor-led workshops, record every one, and almost never watch them again.

So we built a pipeline. Claude Code reads the transcript and proposes chapters. Someone on our team watches the video, checks each cut and signs off, and only then is the clip made. Between 10 July and 5 August 2026 the whole pipeline, including chapters, metadata, thumbnails and shorts, cost us **$47.59 in model spend**. It gave us 37 chapters across four workshop playlists.

People always ask about the $47.59. The part that matters is the sign-off. Remove it and you get slop, just faster.

The second problem was harder to admit. Our courses are written by people who know the subject deeply, which is exactly why they cannot see its gaps. Someone outside Enqurious ran their own GenAI course past a student agent and a teacher agent. The pair found 66 problems, including a quiz on "prompt injection", a term the course never taught. We are building the same pair for our content. A Learner agent reads every section cold, and an Editor agent decides which confusion is a defect and which is productive struggle.

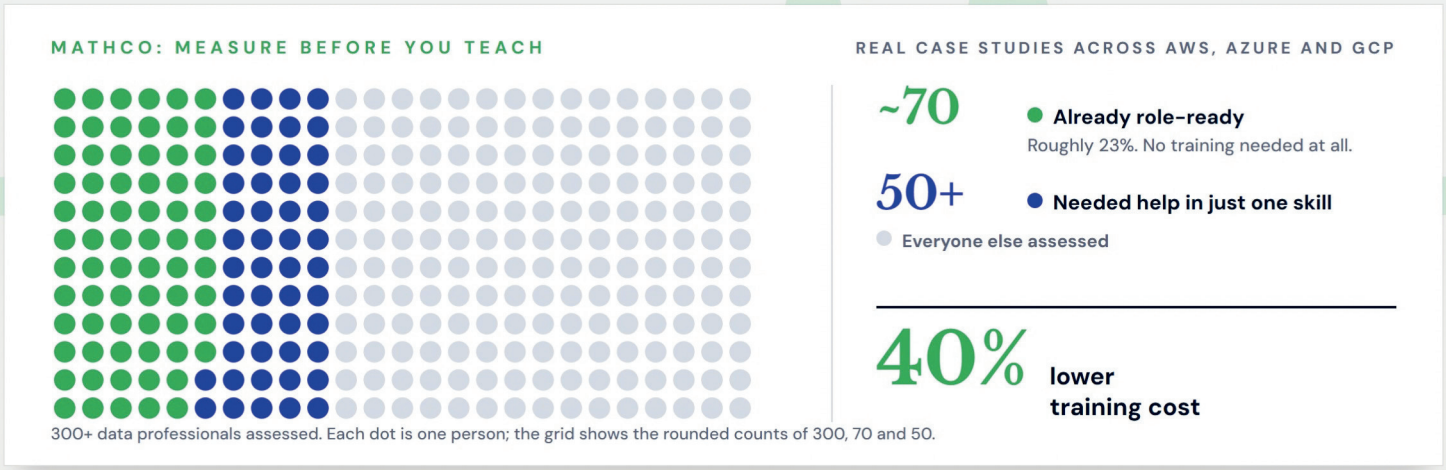


We also built Curio, an engagement assistant that runs parts of our own operations. We have not sold it to anyone yet. It has to earn its keep with our team first.

Then we went back to clients with better questions

With MathCo, the useful question was not what to train. It was who actually needed training. More than 300 data professionals were assessed on real case studies across AWS, Azure and GCP. About 70 of them, roughly 23%, were already role-ready and needed no training at all. More than 50 needed help in just one skill. Training cost came down by 40%.

That did not come from a better course. It came from measuring before teaching. If it does not move the needle on the P&L, it does not work.



What running it on ourselves taught us

A licence does not make anyone a Forward Deployed Engineer. We have watched licences get handed out and more slop come back for review.

What works is exposure before tools. Put people inside a simulated business problem, with stakeholders who want different things, let them feel the pain, and then teach the technology that removes it. We are building those simulators now.

And some of it is unfinished. We can measure what a person can do, one assessment at a time. We cannot yet tell a client, at a glance, who on their bench is ready tomorrow. The middle of that loop runs today. Both ends do not exist yet.

> **WE'D RATHER SHOW YOU OUTCOME THAN A FANCY SLIDE.** The first client we forward deployed into was ourselves, and it is *still the hardest one to please.*
>
> THE ENQURIOUS TEAM

GET YOUR SUBSCRIPTION TODAY!

WWW.ENQURIOUS.COM/SKILLPULSE



Enqurious

Upskill with Enqurious

AI-powered databricks & Gen AI Training for modern data teams

JOIN THE COMMUNITY

SKILLPULSE

EXCLUSIVE MAGAZINE FOR Data & AI Leaders

www.enqurious.com
+91 77959-03563
sales@enqurious.com
HSR Layout, Bangalore