

When Software Gets a Body.

WHEN SOFTWARE GETS A BODY

Software made failure cheap enough to learn from. Physical AI changes the bargain: once intelligence can act upon the world, advantage belongs to those who make every deployment more reliable, repeatable and economically valuable than the last.

Software gave business one of the great economic gifts of the past half-century: reversibility.

Code could be changed after it shipped, a broken page could reload, a bad recommendation could disappear beneath the next one, and a feature could be released, observed and corrected. Failure was never free, and in safety-critical systems it could be catastrophic, but across much of the software economy the cost of being wrong fell dramatically.

That changed how companies learned.

I began my career writing enterprise software, then worked around engineering systems at Ford before spending much of the next two decades building businesses through cloud, platforms and AI. The technologies changed almost beyond recognition, yet one advantage kept strengthening: the distance between making a mistake and correcting it became shorter, making experimentation cheaper and allowing learning to accelerate.

Physical AI inherits software's intelligence without inheriting all of its economics, because once software acquires a body, the world pushes back.

When probability becomes movement

The excitement around Physical AI gravitates naturally towards capability: whether a machine can recognise an unfamiliar object, manipulate it, understand an instruction or adapt to an environment it has not seen before. These are extraordinary achievements, but they are not yet proof of an extraordinary business.

A warehouse operator does not ultimately buy artificial intelligence; it buys throughput. A manufacturer buys yield, cycle time and productive capacity. Customers pay when intelligence changes an economic outcome reliably enough that they can build an operation around it, which means an intelligent machine can still be commercially useless.

Software accustomed us to probabilistic systems partly because humans could absorb uncertainty at the edges. Physical autonomy transfers more of that uncertainty into the system itself, until eventually a probability distribution has to become movement and movement encounters mass, friction, geometry, wear, latency and people behaving in ways nobody modelled.

The error now has somewhere to go.

In 2025, Waymo notified US regulators of a recall covering 1,212 fifth-generation automated-driving systems because earlier software could cause vehicles to collide with certain chains, gates and similar roadway barriers. The affected vehicles had already

received the software remedy by December 2024. The revealing point is not that an autonomous system encountered an edge case, because every complex system does, but that a software judgement became a physical event.

The code could be changed; the consequence could not necessarily be reversed. A production hour cannot always be replayed, a damaged component cannot be patched remotely and a safety incident cannot be rolled back to the previous release.

Physical AI therefore changes the standard of proof from **Can it do this?** to something much harder: **Can we depend on it doing this?**

The distance between those questions is where much of the value will be created.

The prototype is where the easy questions end

Technology companies have always been vulnerable to confusing technical possibility with commercial proof, and I have watched the pattern recur across cloud, AI, platforms and founder-led ventures. The first successful demonstration changes the atmosphere because something theoretical suddenly works, customers lean forward and capital becomes easier to imagine. The temptation is to believe the hardest problem has been solved when, often, it has merely become visible.

The difficult work begins when capability meets customers it was not designed around, legacy infrastructure, regulation, different markets and operating environments that care little for the elegance of the prototype.

Physical AI magnifies that gap because a robot performing a task once proves possibility, while performing it thousands of times through variation, with declining intervention and improving economics, begins to prove a business.

The International Federation of Robotics recorded 542,076 industrial robot installations worldwide in 2024, more than twice the annual number a decade earlier, while the operational stock reached roughly 4.66 million units. Yet even industrial robot safety standards point towards the harder problem: ISO 10218-2:2025 extends beyond the robot into integration, commissioning, operation, maintenance and decommissioning.

The lesson is larger than the standard itself. A useful machine does not exist in isolation; it exists inside a working system, and the commercial product is ultimately not the machine but its ability to perform reliably inside reality.

The deployment learning curve

That gives us a more revealing way to judge a Physical AI company: instead of asking only whether the machine learns, ask whether the **business learns from every machine it deploys**.

The defining economic curve may be the **deployment learning curve**, by which I mean whether each installation makes subsequent installations faster, cheaper, more reliable and less dependent on human intervention.

I learned versions of this lesson the expensive way. At Microsoft, taking cloud, data and AI into large enterprises required turning dispersed expertise into repeatable products,

commercial motions and operating systems. At Amazon, the problem appeared in another form, as creative and technology work that could easily become bespoke had to become increasingly productised so that revenue could grow without delivery complexity growing alongside it.

The technologies and customers were different, but the trap was the same: customers will pay for custom work long before you know whether you have built a scalable business.

Physical AI makes that danger acute because reality generates endless reasons to customise. Every new environment brings another workflow, integration, layout or edge case that can appear strategically important because a valuable customer stands behind it. Some adaptation is inevitable, but the consequential judgement is whether that work expands the product or merely completes the project.

If deployment twenty requires broadly the same engineering heroics as deployment two, the company has accumulated installations but not necessarily scale. If deployment twenty incorporates what the previous nineteen taught—shorter installation, fewer exceptions, better remote resolution, lower intervention and higher uptime—then the installed base is becoming something much more valuable than a collection of machines.

It is becoming a learning system.

The strongest Physical AI businesses will therefore have two learning curves operating together: the machine learning how to perform the task and the company learning how to make that performance economically repeatable. Ultimately, the second determines the value of the first.

Beware the services company hiding inside the robot

This is where growth can become deceptive, because bookings can rise, machines can ship and revenue can grow while the business beneath those numbers becomes progressively harder to scale.

Behind an expanding installed base, implementation engineers may multiply, field technicians proliferate, remote operators intervene more frequently than expected and customer-specific integrations accumulate. Senior engineers can find themselves permanently attached to exceptions that were supposed to disappear after the pilot, leaving a company that appears to be scaling its product while actually scaling the labour required to make the product work.

Enterprise technology has spent decades discovering that bespoke work is remarkably good at impersonating product-market fit. Physical AI simply makes the disguise more expensive.

The revealing question is therefore what happens to the human and capital burden surrounding each additional machine. Intervention per productive hour should fall as deployment accelerates; installation labour should decline as knowledge becomes repeatable; failures should increasingly be resolved remotely; and the economics of the fleet should strengthen with experience.

If those things are not happening, revenue growth may simply be financing an expanding service obligation, and the P&L eventually discovers what the product story omitted.

Reliability is a growth strategy

Reliability is usually framed defensively, as the discipline that prevents failure, but this understates its economic power because a machine that remains productive longer generates more useful capacity from the same capital. Fewer interventions reduce cost-to-serve, faster recovery protects throughput and greater predictability allows customers to redesign operations around the system rather than preserving human redundancy beside it.

Reliability can therefore improve both sides of the economic equation, giving customers more dependable value while allowing suppliers to deliver it with less incremental effort.

More importantly, it can compound. Greater reliability produces higher utilisation, which creates more operating experience and exposes the failure modes that actually matter; resolving those failures then improves reliability again, turning the fleet from installed hardware into an evidence system.

This matters because many visible ingredients of Physical AI will diffuse. Models will improve, sensors will commoditise, actuators will get better and compute will become cheaper, making capabilities that look exceptional today increasingly available to competitors.

The harder advantage may accumulate in knowing which edge cases actually stop production, which interventions can become remote, which environments destroy the economics, what should remain configurable, which customers should be refused and where human judgement creates more value than removing it.

That knowledge is earned in operation, but its value depends on whether it travels. If every failure teaches only the site where it happened, the company is providing service; if every failure makes the installed base better, the company is building advantage.

Autonomy is not the absence of people

There is a seductive tendency to describe autonomy by how many humans it removes, when the more revealing question is where, how often and at what cost humans still intervene.

A person continuously supervising one machine belongs inside the unit economics of that machine, whereas a specialist capable of resolving rare exceptions across hundreds of machines represents something fundamentally different. Both systems contain humans, but only one may produce meaningful operating leverage.

Automation can therefore hide labour rather than eliminate it, and there is nothing inherently wrong with that. Human judgement may remain the safest and cheapest response to difficult edge cases for years. The mistake is pretending the labour does not exist and valuing the business as though it has already disappeared.

Autonomy should be judged not by whether people vanish from the system, but by whether each intervention becomes rarer, more valuable and more scalable. What matters economically is declining human dependency per unit of useful work.

Recurring revenue can become recurring liability

The same discipline applies to the business model because selling outcomes such as availability, throughput or productive capacity can align a supplier much more closely with what customers actually value. IFR reported 42% growth in robot-as-a-service fleets in professional service robotics in 2024, although traditional sales remained the principal monetisation model.

Outcome-based economics, however, also move risk back towards the supplier. Downtime, maintenance, intervention and under-utilisation can become your economics rather than the customer's, which is attractive when reliability improves faster than the obligations assumed and dangerous when it does not.

A technician repeatedly travelling to restore service is not a footnote to recurring revenue, any more than a remote operator rescuing autonomy or an engineer maintaining a customer-specific integration is. The mistake is to look at the shape of the revenue and assume that the economics have become software-like.

Recurring revenue deserves a premium when the recurring obligation beneath it becomes progressively more efficient. Otherwise the company has not built SaaS with machinery; it has written an insurance policy against its own reliability.

Scale reveals what the prototype concealed

Scale rarely creates complexity from nothing. More often, it reveals complexity that the prototype allowed you to ignore.

I have seen organisations respond to early friction by adding people until another specialist fixes each exception, another layer coordinates the specialists and another process manages the layer. For a while, this works and growth continues, until eventually the organisation itself becomes the workaround for weaknesses in the product.

Physical AI companies are unusually exposed because their complexity crosses software, hardware, safety, integration, field service and customer operations. An engineering exception can become a service problem, a product decision can become a margin problem, a badly qualified customer can become an integration problem, and an organisational delay can become machine downtime.

The operating model therefore becomes part of the product.

What matters is how quickly field reality becomes shared learning: who sees failure data, who can eliminate the underlying cause, whether deployment teams can refuse environments that destroy repeatability, whether sales understands where the product does not yet belong and whether interventions are treated as isolated incidents or evidence that the system itself must change.

A machine can learn rapidly while the company around it learns slowly. Eventually, the customer experiences the slower system.

Seven disciplines for intelligence that has to work

The strongest Physical AI businesses will not necessarily have the machines that appear most intelligent. They will build systems that become more economically dependable each time they encounter reality.

Start with useful work, not artificial intelligence. Define the customer's constraint through throughput, yield, productive time, safety or labour, because intelligence matters commercially only when it changes an outcome consistently enough for the customer to change the operation around it.

Treat deployment as part of the product. Measure the time, labour, capital and exception handling required to reach productive use, then demand that each deployment reduces the burden of the next rather than merely adding another customer.

Expose the hidden humans. Count every intervention required to make autonomy appear autonomous, because remote operators, implementation engineers, maintenance teams and senior engineers resolving exceptions reveal the real maturity and economics of the system.

Make reliability commercial. Uptime, intervention frequency and recovery time belong beside revenue and margin because they determine productive capacity, cost-to-serve and customer confidence.

Know which variation deserves to become product. An unusual customer requirement may reveal a valuable general capability or an expensive distraction, and the judgement is knowing which before bespoke work hardens into architecture.

Price the risk you retain. Outcome-based models create value when reliability improves faster than the obligations assumed, which means recurring revenue becomes more attractive as recurring liability declines.

Make every failure compound. A material failure should improve the machine, deployment process or operating model for everyone who follows, because solving the same class of problem repeatedly means the organisation is paying tuition without learning.

The real race is between learning and complexity

Software's great economic breakthrough was not simply that code could do extraordinary things, but that changing what code did became extraordinarily cheap. That made experimentation easier and learning faster, allowing businesses to discover what worked after the product entered the world and send that learning back into the product almost instantly.

Physical AI inherits that capacity to learn, but it must earn the economics again.

A machine enters an environment it does not control, where it must perform through variation and where failure consumes time, labour and sometimes physical capital. Customers need to trust not merely what it can do in a demonstration, but what it will continue to do on the thousandth cycle when nobody from the founding team is standing beside it.

That changes the investment thesis because the interesting company is not merely the one whose machine becomes smarter. It is the one in which the entire system becomes

more economically certain with every deployment, as installation accelerates, intervention falls, utilisation rises, failures become cheaper to resolve, knowledge travels further and margins strengthen.

The machine has to learn, the organisation has to learn, and ultimately the economics have to learn with them.

That leaves founders, CEOs, boards and investors with a harder and more consequential question than whether the technology works:

Is the business learning faster than reality is adding complexity?

If it is, every deployment makes the system more valuable; if it is not, scale merely gives the problems somewhere larger to live.

Software learned by shipping. Physical AI will have to learn by surviving reality.

Roy Sharples

September 2026

Roy Sharples is the founder of Lucan Ventures, where he builds and reinvents businesses through product innovation, go-to-market and operating model design.

[Lucan Ventures](#) | hello@lucanventures.com

© 2026 Lucan Ventures. All rights reserved.