

Article

Vendor or Team Player?

The difference between delivering what was agreed and caring about what the client actually needs



Prepared by

Vikram Katyani

Founder, IntelliMinds Digital

Executive Summary

What's the difference between being a technology vendor and a true technology partner? Vikram Katyani shares why client outcomes matter more than simply delivering to scope.

I've been thinking about this a lot lately.

What's the actual difference between being a technology vendor and being a technology partner?

They sound like the same thing.

Contents

1. Seeing it from their side
 2. Delivering vs. building something people want to use
 3. But there's a balance
 4. Where I draw the line
 5. When it actually clicks
 6. If I had to sum it up
-

I've been thinking about this a lot lately.

What's the actual difference between being a technology vendor and being a technology partner?

They sound like the same thing.

They're not.

A vendor looks at the contract and asks:

"What did we agree to deliver?"

A team player looks at the client and asks:

"What do they actually need to achieve?"

I've always tried to be the second one.

It isn't always the easiest way to run a business. Sometimes you spend more time than you planned, solve problems just outside the original scope, or take on things you could quite reasonably have called a change request.

But there's something satisfying about looking back at a project and knowing:

"We did the job properly."

Here's the thing: the contract tells you what. The client tells you why.

Every project kicks off the same way — scope, requirements, deliverables, timelines, budget. All of it matters.

But somewhere between the kickoff call and go-live, real life shows up.

Someone's stuck on a process that doesn't quite work.

An admin needs something to behave differently.

A system suddenly has to talk to another system nobody mentioned back in month one.

Or the client realises there's a problem sitting just outside what we originally scoped.

"That's outside scope" is a fair thing to say.

Sometimes it's the right thing to say.

But before I get there, I ask myself something else first:

If I were sitting on their side of the table, what would I want my partner to do right now?

That question has shaped pretty much everything about how I work.

Seeing it from their side

None of this means saying yes to everything, working for free, or pretending commercial reality doesn't exist.

It just means trying to actually understand where the client is coming from.

- Why are they doing this project in the first place?
- What happens the day after go-live?
- Who's actually going to be using this thing?
- What's going to frustrate them?
- What risk haven't they even noticed yet?
- What could turn into a real problem six months down the line?

And the question that matters most:

Would I actually be happy with this if I were them?

That one question alone has stopped me shipping stuff that technically ticked every box but wasn't a job well done.

Delivering vs. building something people want to use

There's a real gap between "we delivered what was specified" and "we built something people actually want to use."

Those aren't always the same thing.

You can satisfy every line of a requirements document and still hand someone a clunky, frustrating process.

Technically, done. Practically, missed the point.

I'd rather catch that while we're building, not after handover. It's the same instinct behind building for usability rather than scope .

Sometimes that means simplifying something nobody asked me to simplify.

Sometimes it's an extra few hours cleaning up a workflow.

And every now and then it means telling a client:

"I don't think we should build it that way."

That's not what people expect to hear from a vendor.

It's exactly what they need from a partner.

But there's a balance

There has to be.

Get close to a client's business and you naturally start answering questions that live just outside the project.

You stay involved when something's not quite right.

Every so often you end up doing work that could have reasonably been a change request.

That's a real risk.

No business survives by giving everything away for free.

Scope matters.

Boundaries matter.

Sometimes the right answer really is:

“That’s outside what we agreed — here’s what it’d cost to add.”

Nothing wrong with that.

Being a team player isn’t the same as being commercially naive.

Where I draw the line

For me it comes down to this:

There’s a difference between protecting your scope and protecting your client’s success.

If something’s a genuinely big ask, we talk about it properly.

But if an extra hour today saves the client a much bigger headache tomorrow, I’m probably just going to spend the hour.

I’ve never enjoyed saying:

“Technically, that’s not our responsibility.”

I’d rather help solve it — partly because it’s the right call, and partly because my name’s on the work too.

When it actually clicks

The best client relationships stop feeling like client-supplier relationships at some point.

You notice the language shift.

It stops being:

“The client wants X.”

And becomes:

“We need to do X.”

The focus moves off who owns the problem and onto how it gets solved.

That’s usually when the good outcomes happen — not because anyone’s ignoring the commercials, but because everyone’s actually pulling the same way.

If I had to sum it up

A good technology partner cares about the outcome as much as the delivery.

- Will this actually get used?

- Will it still make sense in six months?
- Did we solve the client's problem, or just finish our task?

None of that shows up in a statement of work.

It's usually what matters most anyway.

There are projects I look back on and think:

"Yeah, we did more than we priced for."

But if the client's happy, the system's being used, and the actual problem got solved — I don't regret it.

There's something satisfying about knowing what you built is making a difference, not because every box got ticked, but because it was done properly. That is much the same reason we treat getting something into production as the beginning of the commitment rather than the end of it.

So — vendor or team player?

I know which side of the table I'd rather be on.

Not because it's easier.

Not because it's more profitable.

Just because I'd rather build something I'm proud to stand behind than point at a scope document and say "job done."

That's guided how I work for a long time now.

I don't see that changing.

Author Vikram Katyani — Founder, IntelliMinds Digital. Helping organisations move AI from experimentation into production through practical strategy, custom development and managed AI operations.

Continue the Conversation

IntelliMinds Digital helps organisations turn AI ambition into dependable business capability. If this guide has raised questions about your own AI programme, we'd welcome the conversation.

Next steps

- **Take the free AI Readiness Assessment** — a five-minute diagnostic with a personalised report.
- **Book a 30-minute discovery call** to discuss your priorities.
- **Explore the full library** of AI Executive Guides at intellimindsdigital.com/insights/



Read the latest version online

<https://intellimindsdigital.com/insights/vendor-or-team-player/>

Scan to open the live article, which is kept up to date as our thinking evolves.

IntelliMinds Digital

Founder: Vikram Katyani

vikram@intellimindsdigital.com

www.intellimindsdigital.com