# The AIDA Manifesto

## Why Your AI Program Needs a Conductor, Not a Committee

### Where AI & Data Move from Promise to Impact

**AIDA = AI & Data as a Product**
**Draft v0.2 - February 2026**
**By JP Schoeffel**

---

## Table of Contents



---

## How to Read This

This is a manifesto. It is opinionated on purpose.

If you want a neutral survey of AI trends, you will not find it here.
If you want a vendor short-list, you will not find it here.
If you want an implementation manual with step-by-step checklists, you will not find it here.

You *will* find a simple claim:

**Enterprise AI and Data fail at scale because we run them like procurement programs. They succeed when we run them like products.**

A quick note on the name: yes, "AIDA" is also a marketing acronym. This manifesto is not about funnels. It is about ownership. We use AIDA because it is memorable, it signals that AI and Data belong together, and it sounds like what it is: a call to action.

Read the Declaration first. If it makes you uncomfortable, good. That discomfort is a signal that your organization is optimized for process, not for value.

Then read the chapters in order. Each one answers a practical question:

- Why are AI and Data inseparable?
- Why do most AI programs produce no measurable impact?
- Why does "transformation" keep happening without results?
- What did Agile and product thinking already prove?
- What does "AI & Data as a product" actually mean?
- How do we go from problem to proof - fast?
- Who owns this work inside the company?
- How do we govern without killing speed?
- How do we measure what matters?

If you only have 10 minutes, read the Executive Summary and the Declaration.
If you have 60 minutes, read Chapters 1, 4, and 5.

---

## Executive Summary

Most enterprises are not failing at AI because the models are weak.

They are failing because they are using a 1990s playbook to build a 2020s capability.

The playbook looks familiar:

- Declare an "AI strategy"
- Collect requirements from every stakeholder
- Issue an RFP
- Pick a platform
- Hire consultants
- Spend 12-18 months integrating, customizing, and rolling out
- Celebrate "go-live"
- Discover that nothing meaningful changed

You can ship something technically impressive and still deliver zero value.
In many organizations, that is exactly what happens - again and again.

Across industry surveys, the pattern is consistent: **the overwhelming majority of AI pilots never generate (or even measure) meaningful ROI.** The exact percentage varies by report and by definition, but the conclusion does not: the system is producing activity, not impact.

AIDA is a response to that system.

**AIDA stands for AI & Data as a Product.** It makes one move:

> Stop treating AI and Data as a project to deliver.
> Start treating AI and Data as a product to own.

This is not semantics. The moment you treat AI and Data as a product, different questions become natural:

- *What problem are we solving?* (instead of *What are the requirements?*)
- *What is the fastest experiment that could prove value?* (instead of *Which vendor should we select?*)
- *What value did we create in the last 30 days?* (instead of *Are we on schedule?*)
- *What capability did we build internally?* (instead of *Which consultant is running the platform?*)

AIDA is built on four principles:

1. **Problems over Requirements**

2. **Discovery over Procurement**

3. **Value over Transformation**

4. **Capability over Dependency**

These principles are intentionally uncomfortable because they challenge how most organizations allocate budgets, make decisions, and manage risk.

AIDA also makes a second claim:

**AI and Data are one system.** You cannot separate them and succeed.
Data without AI is inert. AI without data is hallucination.

### Signs you're running the old machine

If any of these feel familiar, you are not alone - and you are exactly who this manifesto is for:

- Your main AI artifact is a roadmap.
- Your main AI meeting is a steering committee.
- Your main AI output is a pilot that never reaches production.
- Your main data achievement is a platform that nobody uses.
- Your main metric is budget spent, not value delivered.
- Your best people are tired, and your skeptics are winning.

### What you do on Monday morning

AIDA gives you a practical operating model:

- Pick **one** problem that matters.
- Run a **4-week discovery sprint** to prove (or disprove) that AI and Data can solve it.
- Demand a **Proof of Value** - not a Proof of Concept.
- If the evidence is strong, scale with a product team that owns the full lifecycle.
- Govern with **guardrails, not gates**.
- Measure success with a simple scorecard: value delivered, time-to-value, learning velocity, and dependency reduction.

If you do this, two things happen fast:

1. You stop spending money on AI theatre.
2. You build the capability to turn AI and Data into a compounding advantage.

The goal of this manifesto is simple:

**Make "process-driven AI program" sound as outdated in 2036 as "waterfall software development" sounds today.**

---

## The Declaration

This manifesto was written by a practitioner - someone who has spent decades building products, running systems where reality cannot be negotiated, and leading AI and Data work inside an enterprise.

We are done pretending that AI impact will emerge from committees, roadmaps, and procurement cycles.

We choose a different approach.

### The AIDA Principles

**1. Problems over Requirements**
We value **understanding the problem** over collecting requirements.

**2. Discovery over Procurement**
We value **rapid discovery** over vendor selection processes.

**3. Value over Transformation**
We value **delivering measurable value** over running transformation programs.

**4. Capability over Dependency**
We value **building internal capability** over creating consultant dependency.

> **AI and Data are too important to be run as an IT purchase.**
> **They must be owned as products that earn their place through impact.**

---

# Chapter 0 - One System, Not Two

## Why AI and Data Belong Together

It's Monday morning. You walk into a meeting.

On one side of the table: "Data."
On the other side: "AI."

They have different leaders. Different budgets. Different roadmaps. Different vendors.
They talk to each other once a quarter, usually because something broke.

And then everyone acts surprised when the AI pilot can't get clean, trusted data.
Or when the data platform has no users because nobody built anything valuable on top of it.

This separation is so normal that it feels inevitable.

It isn't.

It's like separating lungs from bloodstream and asking why the body can't breathe.

### Data without AI is inert

A beautifully governed data warehouse that nobody uses is not a strategy. It's an expensive library with no readers.

Data has potential energy. It *could* create value. But potential energy does nothing on its own.
AI is what converts it to kinetic energy: prediction, automation, insight, action.

If your data program measures success in terabytes stored, tables standardized, and dashboards built, you are measuring plumbing - not outcomes.

### AI without data is hallucination

An AI model without grounding in real, relevant data produces confident nonsense.

It's a brain without senses - capable of reasoning, disconnected from reality.

This is why "deploy AI" initiatives often produce:

- Chatbots that sound smart and make things up
- Models with impressive accuracy in a lab that collapse in production
- Automation that optimizes for the wrong objective because nobody clarified what "better" means

Without high-quality, domain-relevant data, AI becomes theatre. It looks alive. It isn't.

### Together, they are one product

Data is the raw material of intelligence.
AI is the mechanism that turns raw material into value.

You cannot have a strategy for one without a strategy for the other. When you try, you get predictable failure modes:

- Data teams build infrastructure nobody needs for the problems that matter.
- AI teams build models on data they don't understand, don't trust, and can't operate.
- Business teams lose trust because outputs change, drift, or cannot be explained.
- Everyone retreats to their lane and calls it "governance."

AIDA treats them as one discipline, one system, one product.

> **The split between "data strategy" and "AI strategy" is not a structure.**
> **It's a symptom.**

### Why enterprises split them anyway

The reasons are understandable. They are also wrong.

- **History:** Data lived in IT and BI long before AI became a board topic.
- **Vendors:** Data platform vendors and AI platform vendors sell different things and reinforce different teams.
- **Skills:** Data engineering and data science are different skill sets, so organizations build different org charts.
- **Fear:** When nobody feels accountable for value, splitting work creates safe boundaries: "not my problem" becomes a job description.

The result is a continuous handoff chain: data to analytics to AI to operations.
Every handoff is a leak.

Value leaks out. Ownership leaks out. Accountability leaks out.

### What "one system" means in practice

AIDA does not require one monolithic team doing everything.

It requires one accountable product team per problem domain - a team that owns the *end-to-end flow*:

- From data quality and availability
- To model design and deployment
- To integration into real workflows
- To monitoring, iteration, and measurable impact

It also requires one simple shift in language:

Stop saying: “What’s our data strategy?” and “What’s our AI strategy?”
Start saying: **“Which problems should we solve with AI and Data - and who owns them?”**

### The quickest diagnostic

Ask two teams one question each:

- Ask the AI team: “What data would make your work useless if it changed tomorrow?”
- Ask the Data team: “What decision would become better next week because of your work?”

If either team cannot answer, you have two programs. You do not have one capability.

### Monday morning implications

- If your AI team cannot answer “where does our data come from?”, you do not have an AI team. You have a demo team.
- If your data team cannot name the top three decisions or workflows their work improves, you do not have a data strategy. You have a storage strategy.
- If AI and Data report into different priorities, you will build two half-systems that never become one full capability.

AIDA begins here, because everything else depends on it.

---

# Chapter 1 - The Broken Machine

## Why Most AI Programs Produce No Impact

It’s Monday morning again.

The AI steering committee meets for the 14th time. The agenda is identical to the last 13 meetings:

- Vendor status update
- Integration risks
- Governance approvals
- “Change management” plan
- A dashboard with green dots that reassure everyone that progress is happening

Someone asks the question that matters: *“When will we see value?”*

The room goes quiet. Then the answer arrives:

“After Phase 3.”

Phase 3 is nine months away.

This is what failure looks like in a modern enterprise.
Not a crash. Not a scandal. A slow, expensive drift into irrelevance.

### The enterprise AI playbook (and why it fails)

Most AI and Data initiatives follow a process that was designed for buying enterprise software.

It looks rational. It is also structurally incapable of producing learning fast enough to create value.

1. Someone declares “we need AI”
2. Requirements are gathered from everyone
3. An RFP is written
4. Vendors perform a theatre of demos and decks
5. A platform is selected
6. Implementation begins
7. Consultants arrive (and never leave)
8. A “go-live” happens
9. The business shrugs and keeps working the old way

This process optimizes for one thing: **predictability.**
It tries to remove uncertainty by planning harder.

AI does not reward planning.
AI rewards learning.

### The uncomfortable data point

Across industry surveys, most AI pilots do not make it to measurable, sustained business value. Some reports put the number of “no ROI” pilots near total. Others are more conservative. The methodology varies.

But no senior technology leader is shocked by the direction.

You can see it in any large company:

- Ten pilots. One in production. None with clear value.
- A data lake that nobody swims in.
- A model that worked in a demo and died in integration.
- A roadmap that looks beautiful and changes nothing.

If your organization has been “doing AI” for two years and cannot point to a single decision, workflow, or customer experience that is measurably better because of it, you are not unlucky.

You are running the broken machine.

> **A 95% failure rate is not bad luck.**
> **It's a system working exactly as designed. Consider what it is designed to produce: activity, not impact.**

### Six anti-patterns that kill value

These are not edge cases. They are the default behaviors of process-driven organizations:

- **The RFP Trap:** six months selecting a platform before knowing what you need it to do.
- **The Consultant Spiral:** every step requires external help, so capability never becomes internal.
- **The Requirements Cathedral:** a document becomes sacred, and reality becomes inconvenient.
- **The Pilot Graveyard:** pilots proliferate because "learning" is never connected to delivery.
- **The Dashboard Delusion:** beautiful charts replace better decisions.
- **The Data Quality Excuse:** "we can't do AI until the data is perfect" becomes a permanent postponement strategy.

Each anti-pattern has a common root: decisions are made without evidence.

### The real costs aren't on the invoice

The budget waste is visible. The invisible costs are worse.

- **Time:** 18 months spent implementing yesterday's assumptions
- **Talent:** your best people learn helplessness and leave
- **Cynicism:** the organization builds antibodies to the next initiative
- **Opportunity:** competitors ship learning cycles while you ship status reports

The most damaging outcome of a failed AI program is not the money.

It's the belief that "AI doesn't work here."

AI works. The program didn't.

### Why the machine keeps running

Because it is comfortable.

- Procurement processes feel safe.
- Committees distribute responsibility.
- Roadmaps create the illusion of control.
- Consultants provide confidence by flooding the room with words.

The machine protects careers.
It does not create value.

### The two questions that expose the machine

Ask any AI program two questions:

1. **What problem are we solving right now - in one sentence?**
2. **What evidence do we have that we're solving it?**

If you get a paragraph and a roadmap, you have your answer.

### Monday morning implications

- If you cannot name the single problem your AI program is solving right now, pause spending until you can.
- If your program's main artifact is a roadmap, demand an experiment instead.
- If your success metrics are budget and timeline, your system will produce budget and timeline - not outcomes.

Chapter 2 explains why the machine persists under a more flattering name: "transformation."

---

# Chapter 2 - Transformation Is a Treadmill

## Why Catching Up Never Creates Advantage

The word "transformation" is everywhere.

It shows up in strategies, operating models, org charts, and board decks.
It sounds ambitious. It sounds important.

In practice, transformation is often a polite name for one thing:

**Spending money to look modern.**

### Transformation vs. innovation

Here's the simplest distinction:

- **Transformation** is adopting what exists.
- **Innovation** is creating what doesn't.

Transformation is catching up. Innovation is moving forward.

There is nothing wrong with catching up. Many organizations must modernize basic systems to survive.

But catching up is not a strategy for winning. It's a strategy for not losing *yet*.

### Why transformation is seductive

Transformation sells itself because it has all the features executives like:

- Clear deliverables
- Predictable timelines
- Big vendor names
- Slide-friendly roadmaps
- A narrative of progress, regardless of impact

It turns uncertainty into a plan.

It also creates a useful shield: if value doesn't appear, you can always say, "we're still transforming."

Transformation is a way to spend money without admitting uncertainty.

### The treadmill effect

Transformation roadmaps are treadmills.

You run hard. You spend a lot. You generate heat.
You do not move.

By the time you finish year one of your "modern data platform" transformation, year two arrives with a new wave:

- new model capabilities
- new tools and vendors
- new governance requirements
- new threats
- new competitors who learned faster than you

You start transforming again. You never stop. Because the industry selling transformation does not get paid for stopping.

> **Transformation is a program.**
> **Innovation is a habit.**

### The hidden damage: transformation kills optionality

Innovation requires options.

When you invest in discovery, you create options: multiple ways to solve a problem, tested cheaply, with evidence.

When you invest in transformation, you often destroy options:

- you lock into a platform early
- you lock into a vendor operating model

- you lock into a roadmap that becomes politically difficult to change
- you lock into a budget that must be defended, even when evidence says “stop”

The organization becomes less capable of changing course. And AI is the domain where changing course is the whole point.

### Why you can’t transform your way to advantage

Competitive advantage comes from capabilities that others do not have.

Transformation, by definition, is adopting capabilities that others already have. It is table stakes.

If you want to differentiate, you cannot outsource your thinking to a vendor roadmap.
You must build a learning system that can discover, test, and scale solutions faster than your competitors.

That is innovation.

### What AIDA changes

AIDA rejects transformation theatre. It does not reject modernization.

It changes the unit of progress:

- Not “platform implemented”
- Not “users migrated”
- Not “roadmap delivered”

But:

- **problems solved**
- **value delivered**
- **capability built**

Transformation asks: “Are we on track?”
AIDA asks: “Did the track lead anywhere worth going?”

### Monday morning implications

- If your AI strategy is a three-year roadmap, shorten your planning horizon and lengthen your learning loop.
- If your program cannot show value in 4-8 weeks on a real problem, your risk is not too high - it’s too low. You’re making big bets without evidence.
- If every initiative is framed as transformation, challenge your team: “What innovation will this create? What will we do next year that we cannot do today?”

Chapter 3 shows that we already lived through this shift once - in software.

---

# Chapter 3 - The Product Revolution

## What Agile and Product Thinking Already Proved

In 2001, a small group of practitioners wrote the Agile Manifesto.

They were not trying to create an industry. They were trying to stop wasting their lives on failed software projects.

They made a simple move: **value over process.**

And it worked.

### Agile changed more than delivery

Before Agile, large software projects were famous for three outcomes:

- late
- over budget
- wrong

Agile did not magically eliminate failure. But it changed the default behavior of teams:

- shorter cycles
- earlier feedback
- tighter connection to users
- more learning, less guessing

It also exposed an uncomfortable truth:

Most failure was not a coding problem.
It was a decision-making problem.

Then product thinking took it further.

### From "build it right" to "build the right thing"

Product leaders like Marty Cagan, Jeff Patton, and Teresa Torres put language to what strong teams were already doing:

- Don't just deliver features.
- Solve problems.
- Run discovery continuously.
- Trust empowered teams to find solutions.

This shifted organizations from output obsession to outcome obsession.

It also introduced a discipline that most enterprises still lack in AI:

**separating ideas from evidence.**

Product teams are allowed to propose. They are also required to prove.

### The transfer to AI and Data is obvious

And yet, most enterprises did not transfer the lesson.

They run software teams with sprints and standups.
They run AI and Data with committees and RFPs.

They practice Agile where it is familiar, and waterfall where it is frightening.

Why?

Because AI and Data are wrapped in uncertainty, and uncertainty triggers the oldest reflex in corporate life:

**control it with process.**

But AI punishes that reflex.

AI work is not a linear assembly line. It is a learning loop:

- You test a hypothesis.
- The data surprises you.
- The model surprises you.
- The user surprises you.
- Reality surprises you.
- You adjust.

If your governance assumes that the plan is stable, you will either lie to the plan or ignore reality. Often both.

### "But we tried Agile and it didn't work"

Many organizations "did Agile" and got:

- more meetings
- more rituals
- the same top-down roadmaps
- the same output metrics
- the same lack of ownership

They adopted ceremonies without adopting values.

AIDA is a warning: do not repeat that mistake.

If you turn AIDA into a checklist, you will get compliance - not impact.

### AIDA is Agile's successor in this domain

AIDA is not "Agile for AI" as a set of ceremonies.

It is Agile's original spirit, applied to AI and Data programs:

- respond to evidence over following a plan
- create value early over perfecting the roadmap
- build capability over buying solutions
- learn fast over looking certain

AIDA also warns against Agile's failure mode.

Agile became a process industry.
If AIDA becomes a certification industry, it dies.

> **If your AIDA adoption creates more meetings than models, you missed the point.**

### The mapping (without turning it into a framework)

You do not need to copy Agile vocabulary. You need to copy its logic.

- Agile shortened delivery cycles. AIDA shortens learning cycles.
- Product discovery replaced "requirements." AIDA replaces procurement-first with discovery-first.
- Dual-track separated discovery from delivery. AIDA treats discovery as a permanent capability, not a one-time phase.
- Empowered teams owned outcomes. AIDA requires teams to own impact, not activity.

The lesson is simple:

**In uncertainty, the only durable advantage is learning faster.**

### Monday morning implications

- If your AI backlog looks like a feature list, rewrite it as a problem list.
- If stakeholders demand certainty, give them transparency instead: show experiments, evidence, and decisions.
- If your organization hates "Agile language," fine. Keep the values, drop the vocabulary.

Chapter 4 turns the philosophy into a concrete model: AI & Data as a product.

---

# Chapter 4 - AI & Data as a Product

## The Shift That Makes Everything Else Work

When leaders hear "product," they often think "something we sell."

That’s too narrow.

A product is something that exists to solve a problem for a specific user, that evolves continuously, and that is measured by the value it creates.

That definition applies whether the user is a paying customer or an internal team.

### The project model is the root of the dysfunction

Projects have a shape:

- fixed scope
- fixed timeline
- fixed budget
- success measured by delivery against plan

Projects are useful for things that are knowable upfront: building a warehouse, replacing a switch, migrating email.

AI and Data work is not like that.

The highest risk is not “can we build it?”
The highest risk is “is this the right thing to build, and will anyone use it?”

Projects are designed to reduce uncertainty by locking decisions early.
AI requires you to keep decisions open until evidence closes them.

### What changes when you treat AI & Data as a product

A product has a different shape:

- continuous ownership
- evolving roadmap driven by learning
- success measured by outcomes
- accountability anchored in a team, not in a plan

Treat AI & Data as a product and four things become non-negotiable:

1. **A real user** (internal or external)

2. **A real problem** (pain, friction, cost, missed opportunity)

3. **A real metric** (value, not activity)

4. **A real owner** (a team with authority and responsibility)

No user? You have a research project.
No metric? You have a hobby.
No owner? You have a committee.

### Products have lifecycles, not end dates

A project ends. A product evolves.

AI & Data products require ongoing work because reality changes:

- data drifts
- customers change behavior
- regulations change
- competitors react
- models improve
- your own business strategy shifts

If your program treats “go-live” as the finish line, you will deliver a system that decays the moment it meets the real world.

“Done” is a fantasy. Ownership is the only sustainable state.

### The one FROM -> TO table (use it once, here)

| Instead of… | AIDA says… |
|---|---|
| “What are the requirements?” | “What is the problem, and who feels it?” |
| “Which platform should we buy?” | “What experiment proves value fastest?” |
| “Are we on schedule?” | “What value did we deliver, and what did we learn?” |
| “Who approved this?” | “Who owns this outcome?” |
| “When is the project done?” | “How will this product improve over time?” |

### Internal AI & Data products

Internal products are often where the waste is largest, because nobody demands a market test.

Examples of internal AI & Data products:

- A decision-support tool for pricing teams
- A forecasting capability for inventory
- A fraud detection system for claims
- An assistant for customer service agents that reduces handle time
- A data product that reliably exposes a specific domain dataset to multiple teams

The key is not what the tool is. The key is that someone uses it, and it changes an outcome.

A useful internal product has three signals:

- **usage is habitual** (people choose it without being forced)
- **the workflow changes** (decisions or actions become different)
- **the metric moves** (cost, time, quality, revenue)

If you do not have those signals, you have infrastructure.

Infrastructure is sometimes necessary. But infrastructure is not impact.

### Customer-facing AI & Data products

Customer-facing products make the product mindset easier, because customers vote with their feet.

Examples:

- personalization that increases conversion
- proactive support that reduces churn
- new pricing models enabled by prediction
- entirely new services built on data and intelligence

In customer products, AI & Data become differentiation.
In internal products, AI & Data become leverage.

Both require ownership.

### "But we need an enterprise platform"

Maybe. But the platform is not the first decision.

The first decision is: **which problem will we solve, and how will we prove value?**

Select platforms after you have evidence about:

- what kinds of data you actually need
- what latency and availability the workflow requires
- what governance constraints are real (not imagined)
- what operating model your teams can sustain

When you buy the platform first, you buy assumptions.

### "But our data isn't ready"

Your data will never be "ready" in the abstract.

Data readiness is not a property of your enterprise.
It is a property of a specific problem.

The only honest way to learn data readiness is to run discovery on a real problem:

- What data do we need?
- What data do we have?

- How good is it?
- What is the minimum cleaning and instrumentation required to create value?

If data quality is the bottleneck, discovery will reveal it early - and then your data investment will be targeted, not generic.

### Platform is a product too (but not the first product)

Enterprises love building platforms first because platforms feel like “foundations.”

AIDA does not reject platforms. It rejects platforms built without clear product pull.

The right sequence is:

- prove value on one problem
- extract reusable capabilities
- build platform services that accelerate the next problems

Platform should be the byproduct of learning - not the prerequisite for it.

### How this scales by company size

AIDA is not only for tech giants.

- **10-50 people:** one small team can own one AI & Data product that matters. The goal is speed, not perfection.
- **50-500 people:** one dedicated product team can run discovery and delivery, supported by a lightweight platform foundation.
- **500-5,000 people:** multiple product teams own different problem domains, supported by a shared platform team that provides reusable capabilities as products themselves.

The scale changes. The principle does not.

> **The moment you treat AI and Data as a product, “implementation” stops being the goal. Impact becomes the only goal.**

### Monday morning implications

- Name your AI & Data products. If you cannot name them, you cannot own them.
- Assign a single accountable team per product. Not a committee. Not a program. A team.
- Demand outcome metrics and usage metrics before funding scale.

Chapter 5 shows how to go from a problem to proof, fast - without betting the company.

---

# Chapter 5 - From Problem to Proof

## The AIDA Discovery Framework

Most enterprises do not fail because they picked the wrong platform.

They fail because they invested heavily before they had evidence that the problem was solvable and valuable.

Discovery is how you buy evidence.

### The one-problem rule

AIDA starts with one problem.

Not three. Not five. One.

Because focus creates learning. And learning creates speed.

If you start with five problems, you will build:

- a platform
- a backlog
- a steering committee

You will not build impact.

### Step 1 - Problem selection: impact and feasibility (with rigor)

Most organizations do problem selection with sticky notes and optimism.

AIDA does it with three hard questions:

- **Impact:** if we solve this, what measurable value appears?
- **Feasibility:** do we have (or can we quickly obtain) the data, access, and change capacity to deploy this into real work?
- **Truth:** is this actually an AI problem, or a process problem wearing a technology mask?

Sometimes the most valuable discovery is: "AI is not the solution."

### Step 2 - Rapid discovery: the 4-week sprint

AIDA discovery is not a three-month phase. It is a sprint designed to produce a decision.

A simple cadence:

- **Week 1 - Problem deep-dive:** map the workflow, users, decisions, pain points, and baseline.
- **Week 2 - Data assessment:** locate the data, evaluate quality, check access, and identify what is missing.

- **Week 3 - Solution experiments:** run fast experiments to see what could work (off the shelf only, see 2.5 below)
- **Week 4 - Proof of Value:** test the smallest viable solution on real data with a measurable outcome.

### Step 2.5 - Off-the-shelf only (zero customization)

Discovery is not the place to build.

Yet this is where many companies quietly lose the plot.

They say: "We're just exploring."
And then they do the most expensive thing possible:

- assemble a team of developers
- hire external consultants
- spend weeks writing requirements
- custom-build a prototype

That is not discovery.

That is a build disguised as learning.

AIDA sets a hard rule for discovery sprints:

- **Use off-the-shelf tools.**
- **Zero customization.**
- **Configuration is fine. Custom development is not.**

You're not trying to win an architecture award.

You're trying to answer one question: *does this change a metric that matters in a real workflow?*

If you can't prove value with what's already available, one of three things is true:

- you're solving the wrong problem
- you're solving the right problem in the wrong way
- your data/workflow isn't ready for AI and should be fixed first

This rule matters because it forces the right behavior:

- it keeps time-to-evidence in weeks, not quarters
- it prevents "prototype debt" that becomes production by accident
- it stops you from outsourcing the thinking before you even know what you need

The alternative is familiar.

When we don't know better, we rely on external teams.

They don't truly know our context, our constraints, our data, or our workflows.

So they compensate with process: workshops, documents, steering meetings, and bespoke code.

Weeks later, we have a demo.

Then we have the bill: tens of thousands for a single use case. ($20k–$50k is not unusual.)

And the worst part: we can’t scale it.

Because it’s custom.

Because it was built around one dataset, one team, one moment in time.

And by the time it “works,” a new tool does the same thing out of the box.

This needs to stop.

Discovery should be a buying decision: **go, kill, or pivot.**

Not a building decision: “can we engineer a one-off?”

Use what exists. Prove value. Earn the right to engineer.

This is not a Proof of Concept.

A Proof of Concept proves technology works.

A **Proof of Value** proves the technology changes a metric that matters.

This is not a Proof of Concept.

A Proof of Concept proves technology works.

A **Proof of Value** proves the technology changes a metric that matters.

## Minimum Viable AI (what it really means)

Minimum Viable AI is not “a small model.”

It is the smallest end-to-end solution that can create a measurable change in a real workflow.

Sometimes that solution includes AI. Sometimes it is mostly data and process.

Minimum viable does not mean low quality.
It means tight scope, fast feedback, and evidence-first decisions.

## A concrete walkthrough: claims leakage in an insurer

Imagine a mid-size insurer. Claims handlers review thousands of claims a week. Leakage happens: incorrect payouts, missed fraud patterns, manual errors, inconsistent decisions.

The legacy approach looks like this:

- Launch a “fraud AI program”
- Buy a platform
- Build a data lake
- Run pilots
- Two years later: a dashboard with “risk scores” that nobody trusts

AIDA runs it differently.

**Week 1:**
Sit with claims handlers. Map the workflow. Identify where leakage occurs. Quantify baseline: payout variance, rework rate, fraud recovery, cycle time.

**Week 2:**
Find the data: claim forms, notes, outcomes, payment histories, policy details. Test access. Identify what is usable *now*.

**Week 3:**
Run cheap experiments: anomaly detection, similarity search, rules on known red flags, and a small model trained on historical outcomes.

**Week 4:**
Deploy a thin “walking skeleton” into the handlers’ workflow for a limited slice: one claim type, one region, one team.

Measure: did payout accuracy improve? Did rework drop? Did cycle time change? Did handlers use it?

Now you have evidence.

Not a roadmap. Not a deck. Evidence.

### A second walkthrough: inventory forecasting in retail

Or take a mid-size retailer with chronic stockouts and overstocks.

The transformation approach builds a “data foundation” for a year and then buys a demand planning suite.

AIDA starts in the aisle.

- Choose one category with high margin impact and frequent stockouts.
- Baseline the current situation: stockout rate, spoilage, lost sales.
- Assess data quickly: sales history, promotions, weather, supplier lead times.
- Run experiments: simple forecasting, then incremental improvements.
- Put a recommendation into the planners’ tool for one category only.
- Measure impact in weeks.

You do not need perfect forecasts to create value.
You need better decisions and faster feedback.

### Step 3 - Decision gate: go, kill, or pivot

At the end of discovery, you decide based on evidence.

- **Go:** the problem matters, the data is sufficient, and the proof shows measurable value.
- **Kill:** the data is missing, the impact is too small, or simpler non-AI fixes work better.
- **Pivot:** adjust the problem framing, scope, or approach based on what you learned.

The kill decision is a success. It saved you 18 months of waste.

### The Evidence Ladder

AIDA uses a simple ladder to prevent premature scaling:

- **Level 0 - Opinion:** “We believe this problem exists.”
- **Level 1 - Baseline:** “We can quantify the problem.”
- **Level 2 - Experiment:** “We have evidence an AI/Data approach can address it.”
- **Level 3 - Production:** “We proved measurable value in a real workflow.”
- **Level 4 - Impact:** “We scaled and the value compounds.”

Most enterprises start spending big money at Level 0.

AIDA does not invest beyond a discovery sprint until Level 2.

### Kill criteria (be explicit)

You should walk away when:

- the data does not exist and will not exist soon
- the workflow cannot change (for legal, operational, or human reasons)
- the impact is too small to justify complexity
- a simple rules/process fix delivers most of the value
- no user wants the solution enough to change behavior

Walking away is not failure. It is discipline.

### Scale criteria (earn the right)

You should scale when:

- the proof of value moved a real metric
- the solution fits naturally into a workflow
- adoption happened with minimal pushing
- you can operate and monitor it with internal capability
- you understand the failure modes and the fallback

**A discovery sprint is not a cost.**
**It’s insurance against 18 months of expensive guessing.**

### Monday morning implications

- Pick one problem and write it in one sentence. If you need three paragraphs, you do not understand it yet.
- Demand a measurable baseline before you fund a solution.
- Run discovery fast enough that you can be wrong cheaply.
- Let's rephrase this: Run discovery with off-the-shelf tools and zero customization. If you need custom build work to prove value, reframe or kill the use case.

Chapter 6 answers the question leaders always ask next: "Who does this work?"

---

# Chapter 6 - Teams, Not Vendors

## Stop Outsourcing Your Intelligence

There is a sentence you should never allow in your company:

"We can't change that because the vendor owns it."

The moment that sentence becomes normal, you outsourced your intelligence.

### The consultant dependency trap

It usually starts reasonably.

- You buy a platform.
- The platform requires integration.
- The vendor provides consultants.
- The consultants configure the system.
- Custom needs appear.
- More consultants arrive.
- Over time, the consultants become the only people who understand the system.

Then two things happen:

1. Costs become permanent.
2. Capability stays external.

You are paying someone else to become good at your business.

### The hidden cost: you lose the learning loop

When discovery is outsourced, the company does not learn.

The vendor learns what works in your environment.
Your people learn how to submit tickets.

This is why outsourced AI programs feel like black boxes: the insight never becomes internal muscle.

### AIDA's team model: small, cross-functional, outcome-owned

AIDA does not require huge AI teams.

It requires empowered teams that own outcomes end-to-end.

A minimal AI & Data product team has:

- **Product leadership:** someone accountable for the problem and the metric
- **Domain expertise:** someone who understands the workflow deeply
- **Data/engineering capability:** someone who can access, shape, and operationalize data
- **Modeling capability:** someone who can experiment with AI approaches and evaluate them
- **Delivery capability:** someone who can integrate into real systems and ship

Sometimes one person plays multiple roles. Sometimes roles expand. The point is not the org chart. The point is ownership.

### Avoid the "Center of Excellence" trap

Many enterprises respond to AI by creating a centralized AI Center of Excellence.

The intention is good: avoid duplication, create standards, build expertise.

The usual outcome is predictable:

- the CoE becomes a queue
- business units become requesters, not owners
- delivery slows
- accountability blurs
- the CoE gets blamed for everything

AIDA prefers a different split:

- **Product teams** own outcomes in domains.
- **Enabling teams** coach, accelerate, and build shared patterns.
- **Platform teams** provide reusable capabilities as products.

Centralization is useful for guardrails and enablement.
Value creation must live where the problem lives.

## The solution can be right. Still wrong for you.

Vendors will often be right about the *problem*. Sometimes they even have a solution that could genuinely solve it.

And yet: it can still be the wrong match for *your* company.

Not because the technology is bad.
Because **you can't own it**.

A solution you cannot sustain is not a solution — it's a deferred incident. It's a future "key person dependency". It's a dashboard nobody trusts. It's a model nobody dares to retrain. It's a platform that slowly turns into a museum exhibit: expensive, impressive, and irrelevant.

This is where many transformations derail: we evaluate tools on **feature-fit** and forget **fit-to-operate**.

Before you buy anything "strategic", run a simple test:

- **Can we run it without the vendor in the room?** (not day one — but within a realistic time horizon)
- **Do we have the skills to implement it properly?** If not: can we upskill in months, not years?
- **Do we have the operating model to maintain it?** (data ownership, governance, support, monitoring, change management)
- **Do we have budget for the *run*, not just the build?**
- **Will this make our teams stronger — or more dependent?**

If the honest answer is "no", the mature decision is not "let's push harder".

It's: **not now**. Or **not this tool**. Or **not at this level of complexity**.

AIDA is not anti-vendor. It is anti-delusion.

Take one step at a time. Choose solutions your organization can actually absorb. Prefer what you can prove, operate, and scale — even if it looks less sophisticated on a slide.

Because "enterprise-grade" is not what the brochure says.
**Enterprise-grade is what you can sustain.**

## “But we’re too big / complex for this”

The bigger you are, the more you need AIDA.

Size does not make process-driven waste smaller. It makes it catastrophic.

AIDA scales through boundaries:

- clear product ownership
- small teams with autonomy
- shared platforms as enablers, not gates
- portfolio-level governance based on evidence

Complexity is not an excuse to go slower. It is the reason to learn faster.

## What teams look like at different scales

**10-50 people**
You likely cannot hire every specialty. That’s fine.

You need one person who can bridge business and technology, plus targeted external help for narrow problems (security review, model evaluation, data pipeline design).

The key: external help teaches and leaves.

**50-500 people**
A dedicated product team of 2-5 people can deliver real outcomes.

They run discovery, ship a thin solution, iterate, and prove value.

They do not wait for a “central team” to grant permission.

**500-5,000 people**
You need multiple product teams aligned to domains (claims, supply chain, customer service, pricing).

You also need a shared platform team that provides reusable capabilities:

- data access patterns
- security and privacy controls
- monitoring and observability
- deployment pipelines
- shared components

But platform is not a gate. It is a service.

The platform team is measured by how much they accelerate product teams - not by how much they standardize.

### The right role for vendors and consultants

AIDA is not anti-vendor. It is anti-dependency.

The right role for external partners:

- accelerate discovery when internal skills are missing
- coach teams on capabilities they can own
- provide tools that reduce time-to-value

The wrong role:

- owning core knowledge
- driving the roadmap
- becoming the permanent operator of your intelligence

**If your AI strategy depends on consultants, it's their strategy, not yours.**

### Hiring for AIDA

Look for judgment before credentials.

You can teach tools. You cannot easily teach:

- curiosity
- problem framing
- comfort with uncertainty
- obsession with value
- willingness to be wrong and learn fast

The best AI people are not always the ones with the fanciest models.

They are the ones who can turn learning into impact.

### Monday morning implications

- Identify one AI & Data product and name its owner team. If you cannot, stop calling it a product.
- Audit your dependencies: who can change the system without vendor involvement?
- Redefine consultant success: "we are independent" is the deliverable.

Chapter 7 shows how to govern this without turning it into chaos - and without turning it back into bureaucracy.

---

# Chapter 7 - Governance Without Bureaucracy

## Guardrails, Not Gates

Executives have a legitimate fear.

AI can create real harm:

- privacy violations
- biased decisions
- security exposure
- hallucinated outputs presented as truth
- automation that breaks trust

The response in most enterprises is predictable:

Add gates. Add approvals. Add committees.

Then the program slows down until nothing happens - which feels safe, because nothing can go wrong if nothing ships.

That is not governance. It's paralysis.

### Regulation is not the enemy

In regulated industries, the instinct is to treat regulation as the reason you cannot move fast.

Regulation constrains what you can do. It does not require you to be slow.

In fact, slow programs can be *more* dangerous: they make large, irreversible bets without evidence, and they discover risks late.

AIDA's claim is simple:

**Smaller bets, faster feedback, and clearer ownership reduce risk.**

### Why traditional governance kills learning

Traditional governance was designed for large capital decisions.

AI discovery is not a capital decision. It is a learning decision.

If every experiment requires approval, you will run fewer experiments.
If you run fewer experiments, you will learn slower.
If you learn slower, you will lose.

The only "safe" AI program is the one that never ships - and that is not safe. It's a slow surrender.

## The AIDA governance model

AIDA governance has four pillars:

1. **Guardrails, not gates**

2. **Transparency over permission**

3. **Ethics by design**

4. **Portfolio thinking**

### *1) Guardrails, not gates*

Define what teams can do without asking:

- what data classes they can access
- what environments they can deploy to
- what budgets they can spend on discovery
- what risk thresholds require escalation

This is the difference between:

- “You must ask for permission to drive”
- “Here are the rules of the road. Drive.”

Practical examples of guardrails:

- Discovery work can use anonymized or synthetic data without extra approval.
- Any use of personal data requires a defined purpose, minimal scope, and audit trail.
- High-impact decisions (credit, hiring, medical, claims denial) require human oversight and explainability criteria.
- Every model must have a rollback plan and monitoring for drift and failure.

Guardrails make speed safer. Gates make safety slow.

### *2) Transparency over permission*

Replace approval meetings with evidence reviews.

Teams share, regularly:

- what problem they are working on
- what experiments they ran
- what they learned
- what they will do next

Leadership stays informed without becoming a bottleneck.

This also creates accountability without theatre. It's hard to hide behind process when you have to show evidence.

### *3) Ethics by design*

AI ethics is not a checklist at the end.

It is a lens applied throughout discovery and delivery:

- Who could be harmed by this?
- What assumptions does the model encode?
- How do we detect drift, bias, or failure?
- What is the human fallback?
- How do we explain decisions?
- What is the acceptable error, and who defines it?

Ethics becomes part of product quality, not a compliance ceremony.

### *4) Portfolio thinking*

AIDA assumes some experiments will fail.

That is not a bug. It is the cost of learning.

A portfolio of AI & Data products looks like:

- a few in discovery (cheap, fast)
- some in early production (learning in the real world)
- a small number scaled (compounding value)
- some killed (lessons captured)

The job of leadership is to allocate budget across the portfolio based on evidence - not to demand that every bet be certain.

## Handling AI-specific risks without theatre

AIDA does not pretend AI risk is trivial. It makes risk manageable by reducing bet size.

Smaller experiments reduce risk because:

- fewer users are exposed early
- failures are caught faster
- data access is scoped
- controls can be tested in real conditions

The illusion of safety comes from big plans and long timelines.

Real safety comes from fast feedback and tight loops.

**Gates create the illusion of control.**
**Guardrails create real control by enabling learning without chaos.**

### Monday morning implications

- Replace one approval meeting with one evidence review.
- Define discovery budgets that teams can spend without escalating every decision.
- Make ethics a design input, not a release checkbox.
- If you are regulated, define risk tiers so that low-risk experiments can move fast without waiting for high-risk reviews.

Chapter 8 closes the loop: what you measure determines what you get.

---

# Chapter 8 - The AIDA Scorecard

## Measure What Matters, or You Get Theatre

Your organization already has an AI scorecard.

Even if you never wrote one down.

It's in what gets praised.
It's in what gets funded.
It's in what gets promoted.

If you reward milestones, you will get milestones.
If you reward value, you will get value.

### What most enterprises measure (and why it's wrong)

Common metrics in process-driven programs:

- budget spent vs. allocated
- milestones hit vs. planned
- features delivered vs. requested
- vendor SLA compliance
- number of models built
- number of dashboards created

These are activity metrics. They are not impact metrics.

They tell you that the machine is running.
They do not tell you that it is producing anything worth having.

### What AIDA measures instead

AIDA uses a simple scorecard that fits on one page.

It focuses on six dimensions:

1. **Problems solved** (outcomes)
2. **Value delivered** (impact)
3. **Time to value** (speed)
4. **Discovery velocity** (learning)
5. **Capability built** (strength)
6. **Dependency reduction** (independence)

*The AIDA Scorecard (example format)*

| Dimension | Question | Example Measures | Type |
|---|---|---|---|
| Problems solved | Which business problems improved because of AI & Data? | problems moved from baseline to improved; adoption rate in workflow | Lagging |
| Value delivered | What measurable value appeared? | revenue lift, cost reduction, cycle time reduction, quality improvement | Lagging |
| Time to value | How fast did we go from problem to measurable impact? | median days from selection to proof of value; to production | Both |
| Discovery velocity | How quickly are we buying evidence? | experiments per week; time per experiment; % killed early | Leading |
| Capability built | What can we do now that we couldn't do 90 days ago? | reusable data assets; deployment maturity; internal skills | Leading |
| Dependency reduction | Are we becoming more or less reliant on outsiders? | % changes requiring vendors; consulting spend | Leading |

| Dimension | Question | Example Measures | Type |
|---|---|---|---|
| | | trend; internal ownership | |

This scorecard does two things:

- It forces clarity: you cannot hide behind “progress” when value is the metric.
- It creates healthy pressure: if discovery velocity is zero, value will never appear.

### The operating rhythm that makes the scorecard real

A scorecard in a slide deck is still theatre unless it changes decisions.

A simple rhythm:

- Weekly: product teams share experiments, learnings, and next decisions (transparency).
- Monthly: leadership reviews portfolio allocation based on evidence (not promises).
- Quarterly: the organization reports value delivered and capability built (not projects completed).

The goal of this rhythm is not to add meetings. It is to replace useless meetings with useful ones.

### Leading vs. lagging indicators

Leaders often want lagging indicators first: ROI, cost savings, revenue impact.

Those matter. But they arrive later.

The best early signal that AIDA is working is not a big ROI number. It’s a learning system that is moving:

- experiments running
- decisions based on evidence
- problems being killed quickly
- one or two products earning the right to scale

### The metric most programs avoid: time-to-value

Process-driven programs hide time.

They talk about “phases” and “roadmaps” to avoid the simplest question:

*How long until anything changes?*

AIDA makes time visible. Because time is your most limited asset.

**If your AI program cannot show value, the first metric to look at is not accuracy.**
**It's time-to-value.**

#### Monday morning implications

- Write your AI scorecard on one page. If it takes ten pages, it's theatre.
- Add discovery velocity to leadership reporting. If it is low, you are not learning.
- Track dependency like you track budget. Dependency is future cost.

---

## Closing - Sound the Trumpets

### A Call to Action

If you read this manifesto and feel defensive, you are not alone.

Most organizations did not choose process-driven AI because they are foolish. They chose it because it is the default system for spending money safely inside large institutions.

But safe spending is not the same as wise spending.

AI and Data will define the next decades of competitive advantage. Not as buzzwords. As capabilities embedded in products, operations, and decisions.

The question is not whether you will invest.

The question is whether your investment will produce impact - or another cycle of theatre.

#### What happens if you don't change

The cost of process-driven AI is not neutral.

- You will spend money and get cynicism.
- You will burn talent and call it "change fatigue."
- You will implement tools and call it "capability."
- You will wake up to competitors who learned faster than you.

This is how good companies become slow companies.

#### Start small. Start real. Start now.

AIDA does not ask you to reorganize the company tomorrow.

It asks you to do one disciplined thing:

1. Pick one meaningful problem.
2. Run one discovery sprint.
3. Demand one proof of value.

4. Scale only what earns the right to scale.

Do this once and the conversation changes.

Instead of debating vendors, you debate evidence.
Instead of celebrating milestones, you celebrate outcomes.
Instead of buying capability, you build it.

### The vision

In ten years, we should look back on process-driven AI programs the way we look back on waterfall software projects:

- well-documented
- carefully planned
- painfully slow
- predictably disappointing

And we should wonder why we tolerated them for so long.

AIDA is not a framework to sell. It is a way of thinking to adopt.

If you are a CIO, CDO, CTO, CEO, or any leader who is tired of spending money without impact:

Print the Declaration. Put it on the wall. Then prove it with one product.

Because AI and Data do not become real through strategy.

They become real through ownership, evidence, and impact.

---

## About the Author

This manifesto is written from the perspective of a practitioner who has spent 30 years building, shipping, and operating technology in environments where reality does not negotiate.

The author's path crosses disciplines on purpose:

- A **PhD in Quantum Physics**, trained in first-principles rigor
- An **MBA in Strategic Management**, trained in how organizations allocate resources and make decisions
- **Six years in strategy and marketing consulting**, learning exactly how transformation theatre is sold and why it persists
- **Two decades as a technology founder**, building sales and marketing automation products that had to deliver value to survive

- Operating a **hedge fund and an algorithmic trading research company**, where the feedback loop is brutally honest: your model works or it loses money
- Leading **AI and Data in a Swiss enterprise**, applying product thinking to the hardest environment: a large organization optimized for process

The conviction behind AIDA is simple:

Ten years ago, the technology wasn't ready. Five years ago, it was almost there. Now it is.

AI and Data will define the next decades. Most companies will waste the opportunity - not because they lack talent, but because they keep using a procurement mindset to build a learning capability.

This manifesto exists to change that.