Why do 70% initiatives fail to deliver value? Because organizations rush to the tool before understanding the problem (even when using AI)

Why Your AI Projects Keep Failing (And How BA/PM/PMO Mindset Fixes Them)

Save this post, you will want to reference this framework in your next planning session!

Let's start with the facts:

1. The Business Analyst (BA) knows what to build.
2. The Project Manager (PM) knows how to build it.
3. The Project Portfolio Manager and PMO knows why, when, and what it's worth.

In consulting and financial services, this mix of skills turns ideas into reliable software. It should also save money, as it stops scope drift and it turns "AI" from hype into a working product!

Yes, I'm saying it: Most AI transformations' fail because they lack the proper discipline described here. Let me know in the comments if you disagree.

First principles: start from the value, not the tool

A sound software product begins with a simple chain:

1. **Problem:** What is hurting, where, and why; or how to create something new!
2. **Decision:** What choice improves the expected outcome.
3. **Data:** Which facts demonstrates that choice.
4. **Process:** How the decision creates the workflow, the new target operational model.

5. **Controls:** Mitigate risks, audit and compliance, control privacy, and cost.
6. **Measure:** Define the metric that proves value in what is being done.

The Business Analysts holds that chain together, the Project Managers turns it into a delivery plan, and the Portfolio/PMO makes sure the effort is laddered to strategy, staffed, sequenced, and measured.

BA/PM/PMO experience designs for these from day one, not as a rescue plan later.

A quick map of roles to outcomes

I've Delivered 50+ Tech Transformations. Here's What Actually Works.

Three real examples from my work (anonymised)

1) Large asset transfer and operations migration (financial services)

What we faced: A complex cut-over of data, orders, and reconciliations across multiple systems and time-zones.

BA work: Mapped end-to-end flows, defined golden sources, designed hand-offs, and wrote the data and workflow specs for the migration tooling.

PM work: Built the dress-rehearsal plan, parallel-run criteria, and rollback steps. Agreed a hard stop for go/no-go with business, compliance, and tech.

PPM/PMO work: Set the benefits and readiness gates, tracked burn vs benefit, and forced decisions on scope trade-offs.

Result: Clean cut-over, faster settlement, and fewer breaks. The team reused the playbook for later migrations, saving weeks each time.

2) Portfolio governance and delivery control in a global operations function

What we faced: A large change portfolio with competing priorities, opaque costs, and weak predictability.

BA work: Standardised initiative charters and data definitions for benefits and costs.

PM work: Established RAID, resource controls, and change control with QA and release.

PPM/PMO work: Introduced real-time portfolio dashboards and an executive cadence. Sequenced

Role	Decides	Core artefacts	Tangible value	Failure avoided
Business Analyst	What problem, what scope, what "good" looks like	Process maps, data contracts, acceptance criteria, service boundaries	Clear requirements, fewer reworks	Building the wrong thing
Project Manager	How to deliver it	Plan, risks, RAID, dependencies, cut-over design, test plan	Predictable delivery, managed risk	Slipping dates and budgets
Project Portfolio Manager / PMO	Why now, how much, and in what order	Roadmap, OKRs, benefits model, funding gates, capacity model, governance	Focus on value, cross-team alignment	Scattered initiatives, value leakage

funding to strategic outcomes.

Result: Delivery predictability rose by c.25%, and executives made quicker, better trade-offs based on live data.

3) Document-heavy review in a regulated process (consulting/financial context)

What we faced: Thousands of pages per case, slow manual triage, and high variation.

BA work: Broke the task into machine-friendly subtasks: classify, extract, validate, summarise. Defined accuracy thresholds and exception paths.

PM work: Built an evaluation harness (golden set), designed the human-in-the-loop step, and measured throughput and error rates.

PPM/PMO work: Framed the benefits case, managed the rollout across teams, and ensured audit and privacy controls.

Result: Throughput up, error rates down, with clear audit trails and faster decisions.

These outcomes were not accidents. They came from BA clarity, PM discipline, and PMO governance working as one.

Why this background matters even more when using AI

Machine-learning and AI agents add three hard truths:

- **Non-determinism**. The same input can produce different outputs. You need test predictions, datasets, and rollback paths.
- **Context management**. Models forget. You must preserve state outside the model and re-apply it.
- **Cost dynamics**. Inference, latency, and data preparation add a new bill of materials.

Translating this skill into "Applicable AI" and Low-code/AI-assisted Development

My early programming and data work (e.g. SQL) changed for me, how working with AI and how products are shaped: **Think like a builder, not just a planner**.

Pick the smallest solution that works, keep data tight, test and measure everything. Then expand!

Applied to low-code/AI-assisted development, this means:

Define the "atomic tasks" correctly: One agent, one job. e.g., "create a table to store this information," "use the data in the table to do this," "Check that the result meets the criteria."

Specify the interface: "The UI needs to display the values calculated in this format," "Use schemas to represent a set of data."

Control context: Keep a local storage for facts, decisions, and IDs. Use the models for only new information that is needed and can be anonymised.

Design the tests: Provide examples, expected outputs, and pass/fail rules. Track the implications over time.

Model size and cost first: Prefer a compact local model if it meets the metric. Call external APIs only when needed, with just the data that will keep the private context out.

Guardrails: Security handling, prompts with constraints, retrieval boundaries, and human review for high-impact actions.

Measure, analyse, and iterate: Log prompts, latencies, token spend, and errors. Improve what moves the value metric first.

Your organization doesn't need AI if you can't even map your current processes properly.

A simple blueprint everyone can reuse

1. Problem framing: Define: user, pain, decision, success metric.

2. Process and data: Map the steps. Name the official sources. Fix the data points first.

3. Agent design: Define one task per agent. Specify inputs/track outputs.

4. Evaluation: Create a small but sharp dataset. Add further variances or use cases, one at a time.

5. Delivery plan: Cut into small deliverables. Ship often. Demo to the business every sprint.

6. Controls: Add privacy, audit, backups, and rollback from day one.

7. Value tracking: Track live metrics: time saved, errors avoided, revenue protected. Be ready to pivot.

Effort and cost: how a BA/PM/PMO Mindset Reduces Surprises

Effort: Break work into small units, estimate with ranges.

Cost: Count people days, environments, and AI usage (tokens, calls, and latency). **Tie cost to the value metric** (what are you paying for what is being delivered).

Risk: Keep a live RAID. Treat non-determinism and hallucinations as first-class risks with tests and human checkpoints.

Bringing it together

BA keeps the work useful.

PM keeps the work possible.

PMO keeps the work valuable.

Build the smallest thing that provides value. Measure it. Then scale!

What's your experience?

Have you seen AI projects fail for lack of proper governance?

Which role (BA/PM/PMO) is most often missing in your organization?

What's one 'atomic task' you'd apply this framework to tomorrow?

#ProjectManagement #ArtificialIntelligence #BusinessAnalysis #DigitalTransformation #FinancialServices #Consulting

Disclaimer and Disclosure