**Potential Questions and Concerns:**

1. **Why can’t we leverage existing tech resources instead of hiring new ones?**
   * **Response:** “the good resources” are already fully committed to other critical initiatives like PLA and Vasara.
   * **The expertise we need for MDSOR is highly specialized**, **combining deep business insight with advanced technical skills.**

By bringing in dedicated resources, we ensure that MDSOR gets the focused attention it needs without compromising other projects.

1. **How do we know this investment will pay off in the long run?**
   * **Response:** Investing in MDSOR now means we’re building a future-proof platform. This will streamline data processes, reduce inefficiencies, and improve compliance and risk management. In the long run, this reduces costs, accelerates product launches, and ensures that we remain competitive. Delaying this investment could lead to bigger challenges and higher costs down the line.
2. **What if the technology landscape changes? How adaptable will MDSOR be?**
   * **Response:** **One of the key principles in our roadmap is to ensure that MDSOR is built with flexibility and scalability in mind**.

We’ll adopt a **modular architecture that can integrate with new technologies** and evolve as the landscape changes. This ensures that MDSOR remains relevant and continues to add value, no matter how the tech environment shifts.

1. **How will we ensure collaboration between business and technology teams?**
   * **Response:** **Collaboration is at the heart of this initiative**. We’ll have dedicated liaisons from the business side working closely with the tech team to ensure that development is aligned with business needs. **Regular check-ins, joint planning sessions, and continuous feedback loops will ensure seamless integration** and mutual understanding.
2. **Why decoupling**

When it comes to decoupling MDSOR from Vasara, we need a strategic approach that ensures **both systems continue to operate smoothly and efficiently**.

On the Vasara side, the goal is to identify and isolate the core functionalities that MDSOR relies on.

Meanwhile, MDSOR will **build its own ingestion, modeling, validation, and distribution pipelines** — **decoupled from Vasara’s deployment cycles and** **priorities**

It’s about **unlocking speed, ownership, auditability, and agility**

The transformation is not just technical — it’s architectural and operational.  
We’re embracing a **modular, federated data mesh model**:

**Why We Need 4 Specialized Business-Side Resources for MDSOR Buildout**

We’re not simply asking for headcount — we’re asking for **embedded specialists** who will drive the success of MDSOR by bridging front-office, risk, modeling, and data governance needs with scalable architecture

This is not copy-paste build work

:

**🔹1. Most Tech Resources Are Not Set Up for This Kind of Work**

**Building a decoupled, production-grade MDSOR is not an infrastructure clone or code port.**

* The best Vasara developers are already absorbed in PLA and compute priorities.
* Generic or junior tech hires will deliver code, but not scalable, aligned, or reusable components.
* Left alone, we risk building **something that works, but doesn’t scale**, doesn’t onboard quickly, and can’t meet evolving FO/Risk needs

**🔹2. These Roles Are Not Just Technical — They’re *Specialized***

* We are looking for **multi-disciplinary profiles** with a hybrid of business, data, and systems engineering skills — the kind Vasara or a tech-only team wouldn't naturally recruit or retain

**🔹3. Vasara Wasn’t Built by Tech Alone — Neither Should MDSOR Be**

Just as Athena and Fusion were built with **deep business ownership**, we need experts who understand:

* The real front-office use cases
* What risk models demand from factor time series
* What governance expects for audit trail and versioning
* How data flows across risk, finance, and compliance

These are not things that can be reverse-engineered from JIRA tickets.

**🔹4. Responsibilities of These Business-Embedded Engineers**

Each of these four roles will:

* 🧠 **Drive functional design**: Define how risk factors are modeled, transformed, versioned
* 🧩 **Design reusable components**: Normalize onboarding, validation, and QA pipelines
* 📊 **Define schemas and taxonomies**: Build the foundational layer for discoverability and compliance
* 🔁 **Embed bitemporality and lineage**: Architect data lifecycle tracking at the core
* 🛠️ **Prototype tooling**: Create UI, SDKs, and pipeline builders that FO and Risk can own
* 🔗 **Write contract-level specs**: Replace fragile class-level imports with durable APIs and service contracts
* 📓 **Codify the data catalog**: Establish domain-aligned metadata, naming, SLAs

**🔹5. Why Tech Can’t (and Shouldn’t) Do This Alone**

If we let tech run this independently:

* We lose functional clarity
* We risk building something disconnected from FO/Risk needs
* We risk repeating past architecture mistakes

We also can’t **control the quality of who gets staffed** — especially in a pressured hiring cycle.  
By embedding ownership with us, we ensure:

* **Continuity**
* **Architecture stewardship**
* **Business accountability**
* **accountability**

**🔹6. This Is a Cross-Platform Investment — Collaboration Is Not Optional**

MDSOR will serve **many platforms beyond Vasara**: Risk, CleanPnL, model validation, audit, even Finance.  
Success depends on **collaboration across business and tech**, not just tickets handed across teams.

We’re not just asking for more people.  
We’re asking for **co-owners of a platform that will serve the entire firm** — and we are ready to provide them.

Here’s a **clear, executive-ready pitch** you can use for your **resource planning meeting** — whether spoken, sent as an email pre-read, or summarized in slides.

**🔷 PITCH: Resourcing for Advanced Data Analytics Functions**

As we look to expand our Advanced Analytics capabilities — including machine learning, model testing, and analytics engine development — it's critical to recognize what this truly requires from a staffing and prioritization standpoint.

**✅ 1. This Work Is Not Complex — But It Is Deep**

“We're not building rocket science, but we are building durable infrastructure that requires time, understanding, and the right people.”

* Advanced Analytics is not just a model here and there.
* It includes:
  + Testing various modeling frameworks
  + Building reusable analytics pipelines
  + Comparing model behaviors
  + Managing experiment traceability, versioning, and performance tracking
  + Enabling interpretability and explainability for governance

This work requires **dedicated ramp-up**, especially when inheriting prior experimentation or partial implementations.

**🚩 2. Ownership = Inheriting the Learning Curve**

“Moving responsibility is not a handoff — it's a handover of intellectual debt.”

* The new team must **absorb prior decisions, data nuances, and tool dependencies**.
* Without space to learn and time to stabilize, we risk:
  + Rebuilding what's already been done
  + Misaligning with regulatory or FO needs
  + Producing inconsistent or non-defensible analytics

**🧠 3. Not Staffing Properly Introduces Real Risks**

“The cost of delay is not just time — it’s credibility, audit failure, and lost insight.”

* Risks include:
  + Inability to validate model behavior before deployment
  + Missed deadlines for front-office validation cycles
  + Loss of institutional knowledge if transitional ownership is rushed
  + Lack of comparability across models due to inconsistent testing framework

**🚀 4. If This Work Is on the Critical Path — Treat It That Way**

“If this work feeds model release, regulatory submission, or pricing decisions, it deserves critical path treatment.”

* We should staff for *delivery*, not just *support*.
* We need roles like:
  + **ML-savvy data engineers**
  + **Analytics platform architects**
  + **Testing and validation analysts**
  + **Product owners** who understand both modeling and stakeholder needs

**🔁 5. What Good Looks Like**

“With proper staffing, we’ll be able to:”

* Build standardized analytics engines
* Run consistent model testing across asset classes
* Reduce model approval cycle time
* Reuse testing logic across models and desks
* Provide FO and Risk with explainable outcomes

**🟩 Final Message**

"If we want to do this well — not perfectly, just responsibly — we need to **commit time, staff, and attention**.  
It’s not that the work is too hard. It’s that **rushing it, under-resourcing it, or assuming handoffs equal readiness** will delay outcomes and compound technical debt."

Let me know if you want this:

* Reformatted as an email pre-read
* Designed into a 1-pager for execs
* Delivered as a spoken script for your meeting

Ready when you are.