SUPPLY CHAIN RESILIENCE

When AI becomes a supply chain component.

Treating foundational models like any other critical physical vendor.

supply chain · risk management · AI governance · digital twin

> The newsletter for C-suite leaders who prefer treating software risk like hardware failure.

What is actually happening?

We have a fundamental misunderstanding of what we are integrating into our operations today. Many leaders in the manufacturing and supply chain space treat new AI capabilities as if they are mere "software upgrades" or convenient cloud services—like an email system or a basic inventory database.

This is not just a semantic error; it is a structural risk.

When you integrate a foundational model into your production workflow, you aren't just buying a service. You are installing a critical component in the machinery of your business. If a robot on your assembly line relies on a specific sensor to detect a defect, that sensor isn’t "just software." It is a physical dependency. If that sensor fails or its specifications drift, the line stops.

Currently, many organizations treat AI as if it were an optional utility—something you just plug in and let run. They assume because the interface is easy to use, the underlying logic is stable and independent of the provider's whims. This is **The Utility Illusion**. You are treating a core piece of your "manufacturing equipment" (the model) as something that will always be there exactly as it is today.

In reality, these models are often black boxes owned by third parties who can change their weights, update their parameters, or throttle their performance at any moment. If your quality control logic or your logistics routing depends on a specific output from a provider’s model, you haven't just "adopted AI." You have outsourced a critical part of your process to an external vendor whose priorities may not align with yours.

Why this new dependency persists

The reason we fall into this trap is the allure of the "easy button." In any manufacturing environment, there is a constant tension between the desire for rapid improvement and the requirement for long-term stability.

When an AI model provides a massive jump in efficiency—whether it’s predicting maintenance needs or optimizing pallet loads—the immediate gains are so obvious that we skip the standard "rigor" phase of procurement. We see a tool that works today, and we ignore the fact that we don't own the tools under the hood.

The Convenient Rationalization	The Operational Reality
"It’s just an API call; it’s not like we have to maintain the hardware."	If the API changes or fails, your automated workflow has no fallback and stops instantly.
"The model is too advanced for us to build ourselves, so we'll just use theirs."	You are building a house on land you don't own; the landlord can change the rules at any time.
"It’s only being used for 'non-critical' tasks like internal communication."	Any logic that influences your production schedule or quality checks is, by definition, critical.

We allow this dependency to persist because it feels like a shortcut. But in the world of Six Sigma and Kaizen, we know that any process that relies on an external factor you cannot control is not a "process"—it's a vulnerability. We are mistaking the speed of adoption for the stability of the system.

What happens when that model fails?

When a physical component in your supply chain fails—a bearing in a motor, a faulty valve, or a shortage of raw steel—you have a protocol. You have a secondary supplier, a maintenance plan, and an understanding of how to bypass the failure until you can get parts.

If a foundational AI model "fails" or changes significantly, your response is often paralyzed. This isn't just about the system going offline for an hour; it’s about **The Logic Drift**.

Because these models are constantly updated by their providers, the way they interpret data can change overnight without warning. If your automated quality-check script suddenly begins to "hallucinate" or produce different results because of a backend update, you don't have a mechanic to call. You have an invisible shift in the logic gate that determines whether a part is shipped or scrapped.

The costs of this lack of control are high:

1. **Data Stagnation:** Your systems become unable to process information if the primary model becomes unavailable or changes its output format.
2. **Process Paralysis:** A failure at the API level halts automated workflows, forcing your people back into manual workarounds that they haven't been trained for in months.
3. **Strategic Erosion:** You lose the ability to replicate success because you don't actually know *how* the model produced the result; you just know it did.

If a "black box" is making decisions about your inventory, your safety protocols, or your quality standards, and you cannot swap that box for another one in under an hour, you haven't integrated technology—you've surrendered control to a vendor.

Building a resilient digital supply chain

To fix this, we must stop treating AI as "magic" and start treating it like any other critical piece of equipment on the shop floor. If you wouldn't buy a specialized motor for your assembly line without ensuring you could source replacement parts or have an alternative configuration, don't do it with your digital infrastructure.

We need to move toward **Model Portability**. This means building "buffers" between your operations and the raw AI models. You want to be able to swap out one provider's model for another—or a local version of that same model—without rewriting your entire operation.

To achieve this, we must apply three principles:

1. **The Abstraction Layer:** Never plug your core processes directly into a third-party API. Build a "middle" layer in your software that handles the communication. This allows you to swap the underlying engine while keeping the "dashboard" and the logic of your operation intact.
2. **Multi-Model Strategy:** Just as we have secondary suppliers for raw materials, maintain relationships with multiple model providers. If one goes down or changes its behavior, you should be able to flip a switch to an alternative.
3. **The Local Fallback:** For critical "must-run" functions—like safety checks or primary production gates—you must have a local, hosted version of the logic available. This is your "manual override." It might not be as fast or as "smart," but it ensures that if the internet goes out or the vendor changes their mind, the line keeps moving.

Tomorrow's operational checklist

You don't need to overhaul your entire IT department by Monday morning. You do, however, need to start identifying where the "hidden" dependencies are hiding in your current projects.

On your next project review or during your next walk through the systems your team is building, check these five boxes:

1. **Identify the Critical Path:** List every automated process that directly impacts production volume, safety, or quality. If an AI model touches any of those three areas, it is a "Critical

Component."

2. **Create a Model BOM (Bill of Materials):** Document exactly which models are being used for each task. Don't just list the name; note who owns the model and what happens if their service changes or disappears tomorrow.
3. **Audit API Dependencies:** Ask your engineers: "If this specific provider changed their output format by 10% today, would our system break?" If the answer is yes, you need an abstraction layer.
4. **Define the 'Manual Override':** For every automated AI-driven step, define what the human operator does if the data comes back as a "null" or if the service is unavailable. You must have a practiced transition plan to manual mode.
5. **Establish Portability Requirements:** Moving forward, any new software procurement involving AI should require a "Migration Plan." If you can't move your data and logic out of their system within 30 days, don't buy the product.

Treat these models like parts in your inventory. They are tools to help you make better products; they shouldn't be allowed to become the masters of your process flow.

Call to Action

What is the single most critical component—physical or digital—that currently gives you the least confidence? Share your thoughts with us at newsletter@sixsigmakaizen.com and share this issue with a peer who needs to think structurally about their dependencies.

Newsletter replies: newsletter@sixsigmakaizen.com · X: @kaizen_6sigma

Stay curious, stay improving,

David Rodgers
SixSigmaKaizen.com

References: Logistics Viewpoints, Open Models: Strategic Dependencies (2026)