SHRINK SOFTWARE

ARCHITECTURE BRIEF

Why ERPs Fail at the Production Floor.

A structural argument for why NetSuite, SAP, Oracle, and Acumatica cannot capture what FSMA 204 requires, and what to do instead.

PUBLISHED BY

Shrink Software LLC

AUDIENCE

Operations and IT leaders considering whether their existing ERP can be extended to meet FSMA 204 requirements

DATE

May 2026

FORMAT

Opinionated brief — a structural argument intended to inform a specific decision

EXECUTIVE SUMMARY

The ERP question that won't go away.

Every institutional food manufacturer evaluating FSMA 204 software runs into the same internal debate. We already have NetSuite (or SAP, or Oracle, or Acumatica, or Microsoft Dynamics). Won't our ERP handle this? We bought it because it was supposed to be the system of record for the business. If it can't handle FSMA 204, what does that say about the system, and what does it say about the people who chose it?

This document answers the question directly. The answer is structural, not vendor-specific. ERPs are not architecturally designed to capture the kind of data FSMA 204 requires. Some ERPs have FSMA 204 modules; the modules vary in quality but they cannot fix the structural problem because the structural problem is what the ERP is. Operators who try to make their ERP do this job typically spend 12 to 24 months discovering that it cannot, then have to start over.

This brief is shorter and more opinionated than most of what we publish. It exists because the ERP question is the single most expensive decision operators face in their FSMA 204 implementation, and the wrong answer is invisible until the implementation is well underway.

SECTION 1

What ERPs are actually built for.

An ERP is a transactional system of record for the financial and inventory state of the business. It models the business as a set of transactions: a purchase order is a transaction, a receipt is a transaction, an inventory transfer is a transaction, a sale is a transaction. The ERP's job is to keep those transactions consistent, post them to the general ledger, and produce financial statements that are accurate to the dollar.

This is a hard job. ERPs do it well. NetSuite, SAP, Oracle, Acumatica, and Microsoft Dynamics are sophisticated platforms with decades of engineering investment behind them. Operators are not wrong to expect a great deal from them.

But here is the structural reality: **transactions are not workflow events**, and FSMA 204 requires workflow events.

The transaction-versus-event distinction

A transaction is a logical record of a state change in the business. A purchase order moves from "open" to "received." Inventory moves from one location to another. Cash moves from one account to another. The transaction captures what changed, but it does not capture *how the work happened* — who did it, when exactly, in what order, on what equipment, with what corroborating physical observations.

FSMA 204 requires the workflow event detail, not the transaction record. The Traceability Lot Code has to be captured at the moment of receiving — not when the receiving record gets entered into the system at end of shift. The transformation has to be captured at the moment ingredients are used — not when the production order is closed in the ERP. The cooling log has to be timestamped at the moment the cooling cycle runs — not when someone enters it later from memory.

An ERP can store these workflow events. It cannot generate them. The generation has to happen on the production floor, at the moment of work, by the operator doing the work. ERPs are not designed to be on the production floor. They are designed to be in the back office.

SECTION 2

Five structural reasons ERPs fail at this.

Reason 1: The user model is wrong

ERPs are designed for transaction operators — purchasing clerks, AP processors, inventory managers, financial controllers. The user is typically at a desk, in front of a computer monitor, working through a queue of transactions to post. The interface optimizes for keystroke efficiency on dense data entry forms.

The production floor user is different. They are wearing gloves. They are interacting with food. They are moving between physical workstations. They speak whatever language they speak, which may not be English. They cannot stop to log into a desktop application and navigate three menus to record that they just cooled a tray of fish.

When an ERP module is built for production floor use, it is almost always an adaptation of the desktop interface. The adaptation produces a user experience that is technically functional and operationally impossible. Floor staff revert to paper. Paper does not produce FSMA 204 records.

Reason 2: The data model is wrong

ERP data models are built around inventory and financial objects: items, lots, locations, transactions, accounts. They can represent these objects in great detail. What they cannot easily represent is the workflow itself — the sequence of operator actions that produced the inventory change.

Specifically, an ERP can tell you that production order #4847 used 50 lbs of fish lot ABC-225 and produced 412 units of finished product. It cannot easily tell you:

- Which operator authenticated for the work
- Whether the cooling cycle ran for the required duration
- Whether the operator caught and reported a temperature deviation mid-run
- Whether the labels were physically applied at the moment the system recorded label application
- Whether the receiving inspection actually happened or was logged retroactively

FSMA 204 cares about all of these. The workflow events that produce the inventory transaction are the records the rule requires. ERPs treat the inventory transaction as primary and the workflow events as ancillary. The rule treats the workflow events as primary.

Reason 3: Timing is wrong

ERPs are designed for batch posting. Transactions are recorded, reviewed, validated, and posted, often with hours or days of delay between the physical event and the system record. This is correct for financial transactions — you want validation and review before money moves.

It is wrong for workflow events. The FSMA 204 24-hour response rule depends on records that exist in the system at the moment the work happens, not records that get assembled at end of shift or end of day. An ERP-based FSMA 204 implementation tends to produce records that exist in the system but were generated hours after the work occurred. Under FDA audit, the timestamp on the record is the system timestamp, not the work timestamp. The discrepancy is a failure mode the rule is specifically designed to surface.

Reason 4: The integration model is wrong

ERPs are good at integrating with other transactional systems: order management, warehouse management, payment processing, customer relationship management. They expect their integration partners to also be transactional systems with structured APIs and batch data exchange.

The production floor is not a transactional system. It is a physical environment with operators, equipment, ingredients, finished products, and time pressure. The integration challenge between an ERP and the production floor is not a structured-system integration problem — it is a human-and-physical-process problem. ERPs do not have native concepts for this. Vendors who claim their ERP module is "fully integrated with the production floor" usually mean they have an interface that the floor staff are expected to use. Whether floor staff actually use it is a separate question and the answer is usually no.

Reason 5: The change management model is wrong

ERP implementations are large, expensive, and involve significant business process re-engineering. The implementation methodology assumes that you will reshape your business processes to fit the ERP's model, with consulting support. This is appropriate for back-office processes which are amenable to standardization.

Production floor workflows are not amenable to ERP-style standardization. Every facility has slightly different equipment, different staff languages, different cooling cycles for different products, different label formats for different customers. An ERP module that tries to standardize these workflows breaks the operations it is trying to support. A system that adapts to local workflow conditions cannot be an ERP module because ERPs are not architecturally designed to be locally adaptive.

SECTION 3

The architectural comparison.

The differences between an ERP-based approach and an execution-layer approach are not differences of opinion or vendor preference. They are differences of structural design. The two architectures are optimized for different problems and produce different operational outcomes.

DIMENSION	ERP-BASED APPROACH	EXECUTION-LAYER APPROACH
Primary data model	Inventory and financial transactions	Workflow events at the point of work
User	Transaction operators at desks	Production floor staff at workstations
Interface	Dense desktop forms, keyboard-driven	Mobile, scan-driven, multilingual
Timing	Batch posting, often delayed from work	Real-time at moment of work
Source of compliance records	Reconstructed from transactions	Generated as byproduct of work
Audit defensibility	Depends on retroactive record quality	Audit trail is the operational record
Floor adoption	Often parallel paper workflows persist	System is the workflow; no paper
FSMA 204 readiness	Theoretical (records exist but provenance is weak)	Operational (records are the work)

SECTION 4

What to do with the ERP you already have.

The argument above is not "throw out your ERP." Throwing out the ERP is the opposite of what most operators should do. The ERP is doing the financial and inventory work it was bought to do, and that work matters. The argument is that the ERP is the wrong tool for the additional job FSMA 204 is asking you to do, and the right structure is a different tool that sits alongside it.

Keep the ERP. Add the execution layer.

The pattern most institutional food manufacturers converge on is to keep their existing ERP for finance and inventory, and add a dedicated execution-layer platform for floor-level workflow capture. The two systems integrate at a defined boundary: the execution layer captures the work, generates the compliance records, and feeds inventory and financial events to the ERP. The ERP continues to run the books.

This structure has three operational advantages:

- **No replatform.** The financial system is unchanged. The chart of accounts is unchanged. The reporting and forecasting tools are unchanged. The implementation risk is bounded to the new execution layer, not extended across the whole business.
- **Best tool for each job.** The ERP is doing what it is designed to do. The execution layer is doing what it is designed to do. Neither system is being asked to operate outside its design envelope.
- **Bounded integration scope.** The integration between the execution layer and the ERP is well-defined: order-to-invoice flow, inventory state reconciliation, supplier data sync. The integration scope can be engineered correctly because it is a finite set of data flows, not an open-ended request to "make the ERP handle production."

The "we'll customize NetSuite" trap

The most common alternative to this pattern is a heavily customized ERP implementation in which NetSuite (or SAP, or Acumatica) is extended with custom modules, custom workflows, custom mobile interfaces, and custom integrations to attempt to make it handle production floor work. We watched this approach fail directly inside Anu Sushi LLC after an $80,000 NetSuite implementation that was unable to capture what production actually required.

The failure mode is structural: each customization adds technical debt to a system that was not designed for this purpose, and the cumulative debt eventually exceeds what the customization budget can sustain. Operations end up with an ERP that is too customized to upgrade, too brittle to extend further, and still unable to produce the records FSMA 204 requires. The original NetSuite implementation cost becomes sunk; the operational gap remains unresolved.

THE FOUNDING INSIGHT

Shrink Software exists because this happened to our sister company.

Anu Sushi LLC spent $80,000 trying to make NetSuite handle institutional food production. The implementation failed for the structural reasons described in this document. Shrink Manager was built afterward, by the same operators, to do what the ERP could not. The product exists because we lived the wrong answer to the ERP question and had to engineer the right one.

SECTION 5

How to test the question for your operation.

If your team is currently debating whether your existing ERP can handle FSMA 204, the following questions resolve the debate quickly. Each one tests a specific architectural reality the previous sections describe.

Question 1: Can a production floor operator complete a receiving log entry, in their primary language, on a mobile device, in under 60 seconds, without leaving the floor?

If yes, the ERP has been adapted to the production floor in a meaningful way. Investigate further. If no — or if the answer requires a workstation, a keyboard, and English — the ERP is operating in its native mode, which means it is not capturing what the rule requires.

Question 2: When the production order is closed in the ERP, where exactly did the cooling cycle data come from, and what is the timestamp on that data?

If the answer is "the operator entered it when they closed the order," the data was generated at the wrong time. The cooling cycle data needs to exist in the system at the moment the cooling cycle happened, not at the moment the production order was closed.

Question 3: If FDA called you tomorrow and requested all production records for a specific lot from 14 months ago, what would your team actually do?

The honest answer reveals whether your current system can pass the rule's 24-hour response test. If the answer involves searching for paper logs, reconstructing data from multiple sources, or "we'd have to ask production," the ERP is not the source of truth the rule requires.

Question 4: When was the last time the ERP vendor's reference customer base in institutional food production passed an actual FDA traceability inspection?

If the vendor cannot produce reference customers, the ERP module is theoretical with respect to the rule. This is true even when the vendor has many customers — the question is specifically about customers who have passed traceability inspections, which is a much smaller subset.

SECTION 6

About Shrink Software.

Shrink Software builds Shrink Manager, the execution-layer platform that sits alongside NetSuite, SAP, Oracle, Acumatica, and other ERPs to capture the floor-level workflow that those systems were never designed to handle. The platform has been in production for over two years at institutional food sites processing more than one million units per year.

If your team is currently in the "can we extend our ERP" debate, the easiest first step is a 30-minute conversation in which we walk through your specific ERP, your specific production workflows, and where Shrink Manager fits alongside what you already have. If the conversation reveals that your ERP can handle the job and you don't need an additional layer, that is useful information too.

Shrink Software LLC
hello@shrinksoftware.com · shrinksoftware.com