STANDARD
ELECTRIC SUPPLY CO.

# Regaining Control

Modernizing Food & Beverage Manufacturing Without Disruption, Downtime, or Guesswork

# Table of Contents

# Regaining Control

## Modernizing Food & Beverage Manufacturing Without Disruption, Downtime, or Guesswork

*By JM Moss*

# EXECUTIVE SUMMARY

## Why Modernization Matters Before Something Breaks

Modernization is rarely urgent — until it suddenly is.

The plants that manage it best don't wait for failure to force decisions. They pay attention earlier, when systems are still running but confidence is beginning to erode. When predictability declines. When recovery depends more on people than design. When options quietly narrow.

The purpose of modernization is not to eliminate risk, but to understand it — and to regain control over when and how change occurs.[1, 4]

Organizations that approach modernization this way don't move faster. They make better decisions. They avoid being forced into outages, emergency sourcing, and compressed timelines. And they preserve the ability to modernize deliberately, on their own terms.

If this perspective resonates, the next step isn't a project. It's a conversation about where risk is already accumulating — and how much choice remains.

# When a Reliable System Stops Being Predictable

# CHAPTER ONE

For years, the syrup room had been the kind of system no one talked about. It ran continuously, fed every downstream process, and rarely demanded attention. Operators trusted it. Maintenance knew its rhythms. Leadership took its stability as a given.

That silence was earned — but it was also misleading.

From an engineering perspective, the system was no longer as straightforward as it once had been. The control hardware represented several generations of technology layered on top of one another. Firmware had been held in place for years, not because it was ideal, but because changing it risked unexpected interactions. Documentation existed, but it described the system as originally designed, not how it had evolved through years of incremental changes. The people who could explain why certain workarounds existed — not just how to operate them — were becoming harder to replace.

Nothing had failed.

But confidence was starting to erode.

In brownfield Food & Beverage plants, this is how risk usually develops. Systems don't collapse; they drift. A drive is replaced after a failure. An HMI is upgraded to meet a new requirement. A network segment is extended because it's easier than redesigning it. Each decision makes sense in isolation. Over time, though, the system stops behaving like a single, intentional design and starts behaving like a collection of compromises.

What engineers begin to lose in this situation is predictability.

When something goes wrong in a system like this, the challenge is rarely the fix itself. The challenge is understanding the failure well enough to act with confidence. Is the fault isolated or symptomatic?

Will restarting return the system to a known state, or trigger a secondary issue? Is the behavior expected, or is it an interaction no one has seen before?

As long as production keeps running, these questions remain theoretical. The problem is that when they stop being theoretical, the organization often discovers it has fewer options than it assumed.

This is where uptime gets confused with control. A system can meet production targets every day and still be fragile if no one can confidently describe how it will behave under abnormal conditions. Control isn't defined by runtime — it's defined by recoverability.[4, 6]

When supportability declines, recovery quietly shifts from being engineered into the system to being carried by people. Minor faults take longer to resolve. Root cause analysis becomes less conclusive. Over time, even a historically reliable system becomes risky — not because it fails more often, but because the organization can no longer reason about it under stress.

That realization, not an outage, is what finally drew attention to the syrup room.

The response wasn't immediate replacement. It was clarity. The team examined which components limited recovery confidence, where OEM support had thinned, and which interactions were least understood. They mapped the system as it actually existed — controls, drives, networks, and safety — not as it appeared in original drawings.

Modernization became about restoring predictability. By simplifying the architecture, standardizing diagnostics, and moving to platforms designed for long-term support, the syrup room became legible again. Engineers could reason about it. Abnormal behavior could be anticipated. Recovery paths were known instead of guessed at.

## DATA & REALITY CHECK

**$65B+**

**Worth of legacy automation assets currently operating near end of life**

***— ARC Advisory Group***

**10–12 years**

**Typical reliable life of electronic automation components before elevated failure risk**

***— Athena Controls / ARC***

The system didn't become valuable because it was newer.

It became valuable because it was understandable again. This progression—from predictable operation to fragile behavior—follows a common lifecycle pattern observed in aging automation systems ***(Appendix A)***.

## Chapter takeaway

A system becomes risky long before it fails — it becomes risky when the organization can no longer predict how it will behave under stress.[1]

# CHAPTER TWO

## Obsolescence Is an Engineering Problem Long Before It Becomes a Purchasing Problem

By the time the syrup room began drawing attention, the issue was not failure. It was friction.

Replacement parts that once arrived in days now required weeks. Some components were still technically available, but only through secondary channels. Others were available only as refurbished units with uncertain histories. Maintenance responded the way experienced teams always do: they increased spare inventories, delayed changes that might destabilize the system, and relied on workarounds that kept production moving.

None of these decisions were reckless. In fact, they were rational responses to short-term constraints.

The problem was that each response quietly reduced engineering freedom.

As automation systems age, obsolescence reshapes what engineers are allowed to do.[1, 2] Firmware can no longer be updated without fear of breaking compatibility. Hardware replacements must be matched exactly, even when better alternatives exist. Security patches are skipped because they were never validated for platforms that are no longer officially supported.

Over time, the system stops being something engineers can actively improve and becomes something they are careful not to disturb.

This is where obsolescence transitions from a supply chain inconvenience into a technical risk. When components fall outside their supported lifecycle, engineers lose access to testing, validation, and vendor knowledge. Changes that should be routine become high-risk events. The system may still function, but it becomes increasingly brittle, not because it is old, but because it is constrained.

In the syrup room, these constraints were beginning to stack up. Certain components effectively dictated how the rest of the system had to behave. Spare part strategies were driven more by availability than suitability. Decisions were being made to preserve stability today, even if they made tomorrow harder.

That is the point where obsolescence begins to dictate architecture instead of the other way around.

What makes this situation particularly dangerous is that it often goes unnoticed. As long as spare parts can be found and systems can be coaxed back into operation, leadership sees continuity. Engineers, however, see narrowing margins. Each unsupported component becomes a fixed point around which all future decisions must bend.

If this continues unchecked, the consequences are predictable. When a failure finally occurs, the organization discovers that its options are limited. Replacement choices are constrained by compatibility rather than performance. Emergency sourcing introduces risk. Downtime extends not because the fix is difficult, but because the system has backed the team into a corner.

This is why reactive obsolescence management almost always costs more than planned modernization. The expense is not just financial; it is operational. Decisions made under outage conditions are rarely optimal, and they tend to propagate additional complexity into the system.

In the syrup room, this realization changed the conversation. The question stopped being *"Can we still get the parts?"* and became *"What are these parts preventing us from doing?"*

These conditions align with the "constrained" lifecycle stage, where engineering choices narrow and risk accelerates fastest ***(Appendix A)***.

That shift reframed modernization as an engineering problem, not a procurement exercise.

Rather than waiting for components to fail outright, the team identified which elements posed the greatest constraint on future changes and recovery. They evaluated not just availability, but supportability: which components could be validated, secured, and integrated confidently over time. The goal was to reduce dependency on scarce or obsolete hardware and replace it with a platform designed for long-term availability and predictable behavior.

As those constraints were removed, something important happened. Engineering options expanded again. Firmware strategies could be reconsidered. Diagnostics could be standardized. Security and support no longer felt like compromises.

The system didn't just become easier to maintain.

It became easier to *reason about*.

That is the real cost of unmanaged obsolescence — not that parts disappear, but that engineering choices disappear with them.

## DATA & REALITY CHECK

**58%**

**Manufacturers with no formal automation lifecycle management plan**

***— ARC Advisory Group***

**88%**

**Process manufacturers operating automation beyond the OEM's obsolescence date**

***— Rockwell Automation / ARC Advisory Group***

## Chapter takeaway

Obsolescence doesn't force failure; it forces decisions to be made with fewer and worse options.[1, 3]

# CHAPTER THREE

## Why Downtime Rarely Stops Where It Starts

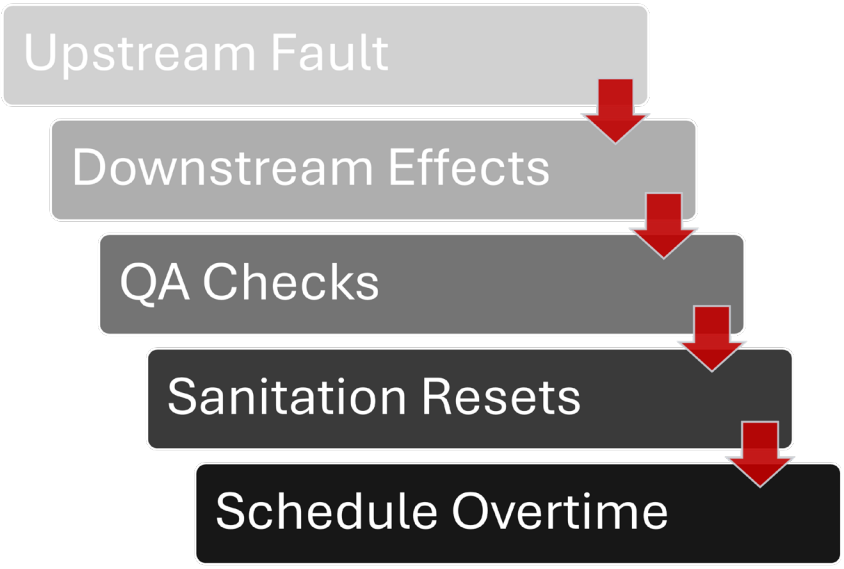
In the syrup room, downtime was never just a mechanical event. The system sat upstream of nearly every critical process in the facility. When it slowed or stopped, the effects were felt immediately — not only in lost throughput, but in sanitation schedules, quality checks, and production sequencing.

That reality is common in Food & Beverage manufacturing. Unlike discrete industries where a single machine failure might affect one cell, process and packaging systems are deeply interconnected. A disruption upstream forces downstream processes into abnormal states, and recovery often requires far more than simply restarting equipment.

Engineers know this intuitively. What is less obvious, especially outside operations, is how much of downtime cost is driven by *recovery behavior*, not failure itself.

In legacy automation environments, failures are often easy to detect but difficult to interpret. Alarms trigger without sufficient context. Diagnostics vary by component and vendor. Restart logic depends on sequence timing, operator intervention, and tribal knowledge. The longer a system has evolved without architectural alignment, the harder it becomes to reason about what will happen when something deviates from normal operation.

As a result, even minor faults can cascade. Product in process may need to be scrapped. Temperature excursions trigger sanitation resets. Quality teams are pulled into investigations that consume hours. Schedules compress, leading to overtime and missed shipments. None of these costs appear on a maintenance work order, but they are very real to the business.



Reducing downtime cost depends less on preventing failure and more on designing systems for predictable recovery ***(Appendix D)***.

## DATA & REALITY CHECK

### $4,000 - $30,000/hr

**Typical cost of unplanned downtime in food and beverage manufacturing, depending on line and product mix**

***— ABB / Manufacturing Digital***

### 5–30×

**True downtime cost compared to visible repair cost when scrap, labor, compliance, and recovery are included**

***— Oxmaint***

In the syrup room, this cascading behavior was one of the clearest signals that modernization was no longer optional. The team recognized that the system was not failing often — but when it did, recovery was increasingly unpredictable. That unpredictability was the true risk.

What makes this particularly dangerous is that organizations often optimize around failure prevention while underinvesting in recovery. Predictive maintenance reduces breakdowns, but it does not eliminate the need to recover safely and quickly when something goes wrong. Recovery speed is often the strongest predictor of downtime cost.[5]

If these conditions persist, downtime begins to reshape behavior. Operators hesitate to restart systems without senior support. Engineers become cautious about making improvements that might destabilize fragile recovery paths. Production leaders pad schedules to absorb variability. Over time, the system becomes slower and more expensive to operate, even if it technically meets uptime targets.

In the syrup room modernization, the goal was not simply to prevent failures, but to **engineer for recovery**. Diagnostics were standardized so faults could be interpreted quickly. System states were clarified so restart behavior was predictable. Monitoring was simplified so engineers could see what mattered, rather than hunting across multiple interfaces.

As recovery confidence improved, something subtle changed: people stopped treating downtime as a crisis. Faults were addressed methodically. Restart decisions were made with data instead of instinct. The cost of each incident dropped, even when failures still occurred.

That is the difference between a system that merely runs and a system designed to recover.

This type of cascading downtime is typical of legacy brownfield architectures with poor state visibility and recovery alignment ***(Appendix B)***.

## Chapter takeaway

Downtime cost is driven less by how often systems fail and more by how well they recover.[5, 6]

# When Compliance Depends on Memory Instead of Systems

CHAPTER FOUR

The syrup room modernization was not initially framed as a compliance initiative. Yet as engineers examined the system more closely, it became clear that automation, traceability, and compliance were tightly linked.

Like many Food & Beverage facilities, the plant met regulatory requirements — but often through manual effort. Data existed across multiple systems. Batch records were assembled after the fact. During audits or investigations, teams relied on spreadsheets, logs, and individual expertise to reconstruct what had happened.

This approach works when time is available. It fails when time matters.

Modern food safety regulations, particularly the FDA's Food Traceability Rule under FSMA Section 204, are built around speed. Manufacturers must be able to provide complete traceability records within **24 hours of a request**.[7]

While the rule does not mandate specific technology, it assumes that systems are capable of capturing and linking data consistently across operations.

Legacy automation environments often make this difficult without tighter system integration.[11] Control systems capture process data, but not in a way that aligns cleanly with traceability requirements. Transformations — mixing, batching, rework — are especially challenging to document accurately without tight system integration. The more manual the process, the more fragile compliance becomes.

In the syrup room, engineers realized that while compliance had not failed, it depended heavily on people bridging gaps between systems. That dependency represented risk — not only during audits, but during recalls or quality events when speed and accuracy are critical.

If these conditions persist, compliance risk compounds quietly. Recall response times lengthen. Data inconsistencies create doubt during investigations. Audit findings consume engineering and quality resources that could otherwise be focused on improvement. Perhaps most importantly, leadership loses confidence in the organization's ability to respond decisively under scrutiny.

The syrup room modernization addressed this risk indirectly but effectively. By improving system integration and data visibility, the plant reduced reliance on manual reconciliation. Batch and process data became easier to access and interpret. Engineers and quality teams could trace material movement and transformations with greater confidence, without assembling information from disparate sources under pressure.

Compliance did not become easier because regulations changed.

It became easier because systems did.

Architectures that rely on manual reconciliation rather than integrated system data increase compliance and recall risk ***(Appendix C)***.

## DATA & REALITY CHECK

### 24 hours

**Time allowed to provide traceability records to the FDA under FSMA Section 204**

***— FDA Food Traceability Rule***

### Manual records common

**Industry studies show traceability data is still largely paper-based or spreadsheet-based across much of the supply chain**

***— FDA / Reagan Udall Foundation***

## Chapter takeaway

Compliance fails quietly when it depends on people instead of systems.[7, 8]

# When Cyber Risk Stops Looking Like an IT Problem

CHAPTER FIVE

No one set out to make the syrup room vulnerable.

Connectivity crept in the way it does in most plants. Remote access was added to support troubleshooting. Data connections were extended so production metrics could be shared upstream. Network segments were bridged because it was faster than redesigning them. Each change solved a real problem at the time, and none of them felt particularly risky in isolation.

The issue was not intent. It was accumulation.

Legacy control systems were never designed for the environments they now operate in. PLCs, drives, and HMIs that were once isolated behind physical barriers found themselves connected to broader networks, often without the segmentation, authentication, or monitoring expected in modern architectures. Firmware that could not be patched remained in place because updates had never been validated for systems already outside their formal support window.

From an engineering standpoint, this created a new kind of uncertainty.

When a fault occurred, teams could no longer assume it originated from a mechanical or electrical cause. A system might behave erratically without obvious physical symptoms. Communications could drop intermittently. Interfaces could become unresponsive without triggering clear alarms. In these moments, the line between "automation issue" and "cyber issue" blurred — and recovery slowed as teams tried to determine what they were actually dealing with.

This is where cyber risk becomes an operational problem.

In manufacturing, a cyber incident rarely announces itself as an intrusion. It presents as loss of control. A system that cannot be trusted to reflect its true state forces conservative responses. Lines are stopped. Restarts are delayed. Engineers hesitate to make changes until they understand whether the behavior they are seeing is persistent, transient, or externally influenced.

The cost of this uncertainty is not measured in data loss. It is measured in downtime.

In the syrup room, this realization emerged during a

## DATA & REALITY CHECK

### 82%

**Organizations impacted by at least one unplanned outage involving machinery/assets over a three-year period [12]**

***— ServiceMax / Vanson Bourne***

### OT Segmentation Gaps

**Insufficient network segmentation can allow threats to move laterally from IT to OT environments and between OT systems [14]**

***— CISA OT Cybersecurity Guidance***

broader review of system behavior. Engineers recognized that increasing connectivity had expanded the system's attack surface while simultaneously reducing their ability to patch, segment, or monitor effectively. Flat network structures meant that disturbances — whether malicious or accidental — had the potential to propagate farther than intended.

Left unaddressed, this type of architecture creates a fragile operating environment. Even routine maintenance becomes risky when engineers cannot confidently isolate systems or validate that changes will not have unintended side effects. Over time, teams compensate by avoiding change altogether, which further entrenches outdated platforms and compounds exposure.

What makes this risk particularly difficult to manage is that traditional reliability metrics don't capture it well. Mean time between failures may remain acceptable. Preventive maintenance schedules may be followed. Yet recovery confidence erodes because the system's behavior under abnormal conditions becomes harder to predict.

If this trajectory continues, organizations face a familiar pattern. Cyber incidents — real or suspected — trigger conservative shutdowns. Engineers lose trust in system state. Recovery requires manual verification across multiple layers. Downtime lengthens not because systems are irreparably damaged, but because no one can confidently say when it is safe to proceed.

In the syrup room modernization, cyber risk was not addressed with standalone security tools. It was addressed architecturally.

By simplifying the system, segmenting networks, and moving to platforms designed for long-term support and connectivity, engineers regained control over system boundaries. Faults could be isolated. Communications paths were clearer. Changes could be validated without fear of cascading effects. The system became not just more secure, but more predictable.

That predictability is what ultimately reduced risk.

Cyber resilience, in this context, was not about preventing every possible intrusion. It was about ensuring that when something abnormal occurred, engineers could understand it, contain it, and recover without shutting down the plant out of caution.

That is when cyber stops being an IT problem and becomes what it really is in manufacturing: a control system problem with production consequences.

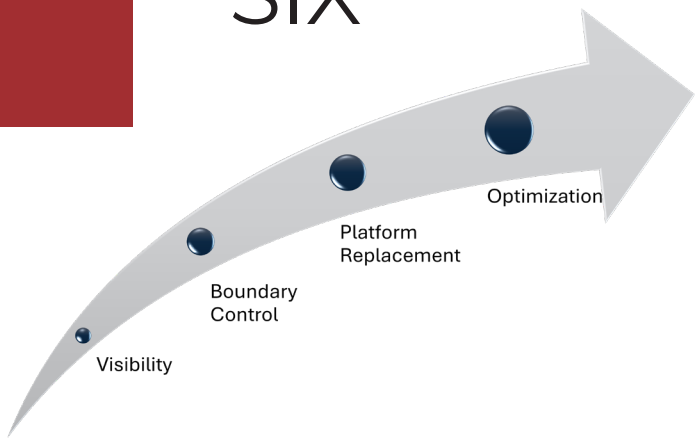
In flat or minimally segmented architectures, cyber events and control faults often present identically at the operator level ***(Appendix B)***.

## Chapter takeaway

Cyber risk becomes expensive not when systems are compromised, but when engineers lose confidence in their ability to control and recover them.[4]

# CHAPTER SIX

## Modernization Without Shutting the Plant Down




The hesitation around modernizing the syrup room was not philosophical. It was practical.

The system worked. It fed critical processes. Any extended shutdown would ripple across production schedules, sanitation cycles, and customer commitments. Engineers knew that replacing everything at once was theoretically clean — and operationally unrealistic.

This is the reality in most Food & Beverage plants. Phased brownfield retrofits consistently reduce unplanned downtime.[10] Production runs continuously, margins are tight, and seasonal demand often leaves no obvious window for disruption. Under these conditions, modernization is often postponed not because the risks are misunderstood, but because the perceived cure feels worse than the disease.

What changes the equation is recognizing that **doing nothing is not a neutral choice**.

As systems age, the cost of intervention does not remain flat. Options narrow. Recovery confidence declines. Changes that could have been staged become emergency responses. Modernization shifts from being planned work to being outage work, and outage work is always more expensive, more disruptive, and less controlled.

In the syrup room, leadership and engineering reached a shared conclusion: if modernization was going to happen — and it would, eventually — it needed to happen on their terms. That meant rethinking what modernization actually looked like.

Instead of viewing it as a single, monolithic project, the team treated modernization as a series of controlled engineering changes designed to reduce risk incrementally. Legacy components that still performed well were allowed to coexist. Interfaces were defined and stabilized before anything was replaced. New systems were introduced in parallel, tested under real operating conditions, and cut over only when behavior was understood and predictable.

This approach demanded more upfront engineering discipline, but it dramatically reduced operational risk. Because the system was never taken offline wholesale, production continued. Because changes were isolated and validated, failures did not cascade. Engineers retained the ability to reason about system behavior at every stage of the transition.

The most important outcome of this approach was not speed — it was confidence.

Modernization without shutdown requires treating uptime as a design constraint, not a byproduct. It requires understanding which parts of the system can tolerate change, which cannot, and how new components will interact with old ones before they ever touch the line. In practice, this often means modernizing around the edges first: diagnostics, networking, visibility, and interfaces — the elements that improve understanding without altering core process behavior.

Over time, as confidence grows, deeper changes become possible.

This is where many modernization efforts fail. Organizations attempt to jump directly from a fragile legacy system to a fully modern one, compressing years of architectural evolution into a single outage. When that fails — or even stumbles — it reinforces the belief that modernization is inherently disruptive.

The syrup room experience demonstrated the opposite. By modernizing in layers, engineers preserved operational continuity while systematically removing the constraints that had made the system brittle. Each change reduced uncertainty rather than introducing it. Each step made the next one easier.

The resulting "after" state reflects a modernized brownfield architecture designed for long-term supportability ***(Appendix C)***.

What this approach ultimately protects is optionality.

When modernization is phased and deliberate, engineers retain choices. If a change introduces unexpected behavior, it can be isolated and corrected. If production priorities shift, the roadmap can adapt. Most importantly, the organization avoids being forced into decisions by failures it did not plan for.

By the end of the syrup room modernization, the plant had not experienced a dramatic before and after moment. There was no grand reopening. Production targets were met throughout the process. From the outside, very little appeared to have changed. From the inside, everything had.

Engineers trusted the system again. Recovery paths were known. Future upgrades felt manageable rather than risky. Modernization stopped being something to fear and became something that could be executed deliberately, one decision at a time.

## DATA & REALITY CHECK

### 20+ years

**Average age of industrial equipment in many U.S. manufacturing facilities**

***— U.S. Bureau of Economic Analysis (via Cognex Brownfield Guide)***

### Months, Not Years

**Brownfield automation projects can often be completed faster than greenfield builds by reusing existing infrastructure [13]**

***— Dematic Brownfield Automation Insights***

### 30%

**Reduction in unplanned downtime reported in phased brownfield retrofit case studies**

***— Xentara / Industry 4.0 retrofit case***

That is the real promise of modernization without shutdown: not faster projects, but fewer surprises.

The syrup room followed a repeatable modernization without shutdown engineering pattern that reduced risk incrementally while production continued ***(Appendix E)***.

## Chapter takeaway

Modernization becomes disruptive only when it is delayed long enough to be forced. [9, 10]

# From Understanding Risk to Acting Intentionally

# CHAPTER SEVEN

If you've read this far, you already recognize the pattern.

Risk accumulates quietly in systems operating beyond their supported lifecycle.[1, 2, 3] Downtime, compliance exposure, and cyber events are rarely the beginning of the story. They are the moment the story becomes visible.

The syrup room mattered not because something went wrong, but because someone noticed **before** it did.

By the time modernization was complete, nothing dramatic had happened. There had been no major outage. No emergency shutdown. No scramble for obsolete parts. Production targets were met throughout the transition. From the outside, the plant looked much the same as it had before.

That was the point.

Modernization is often framed as transformation — a before and after moment defined by new technology and visible change. In reality, the most effective modernization efforts are far less noticeable. They happen early.

They reduce uncertainty before it turns into disruption. And they change how an organization relates to its systems, not just what those systems look like.

In the syrup room, the most important shift was not hardware. It was posture.

Before modernization, the system was something the organization worked around. Engineers were careful not to disturb it. Maintenance relied on experience and memory. Leadership assumed stability because nothing had failed recently. Risk existed, but it was diffuse and difficult to describe.

After modernization, the system became something the organization could reason about again. Boundaries were clear. Recovery behavior — not failure frequency — drove the majority of downtime impact.[5, 6] Obsolescence was visible instead of buried. Change no longer felt like a gamble.

That shift — from reactive problem solving to intentional risk management — is the real outcome to aim for.

Across the industry, many manufacturers understand modernization conceptually, but far fewer approach it deliberately. Systems are upgraded when they break, not when they become fragile. Lifecycle management exists in theory, while technical debt accumulates in practice. Over time, options narrow without anyone explicitly choosing to narrow them.

What the syrup room experience demonstrates is that modernization does not have to be driven by failure to be effective. It can be driven by **loss of confidence** — in predictability, recoverability, and supportability — long before uptime metrics decline.

That distinction changes what comes next.

When modernization is framed as risk reduction rather than innovation, priorities become clearer. Engineers can point to specific constraints instead of generalized concern. Leaders can see how today's decisions affect tomorrow's recovery options. Investment discussions shift from defending cost to protecting control.

Most importantly, the organization retains ownership of timing.

If nothing changes, the pattern is predictable. Systems continue to run — until they don't. Knowledge erodes quietly. Spare part strategies grow brittle. Recovery confidence declines. Eventually, a failure forces decisions that could have been planned years earlier, but are now made under pressure.

The syrup room avoided that outcome by acting while choices still existed.[4]

Modernization did not eliminate risk. It clarified it. Unknown exposure became managed trade-offs. Engineers and leaders gained a shared language for discussing system health beyond uptime and outages.

For teams evaluating where they stand today, a set of practical diagnostic questions is provided ***(Appendix F)***. That is the real end state of modernization: not newer equipment, but organizational confidence.[4]

Plants that reach this state do not modernize less — they modernize more deliberately. They understand which risks matter, which can be tolerated, and which will compound if ignored. Change becomes part of normal operations rather than a response to crisis.

From the outside, they look stable. From the inside, they are prepared.

## DATA & REALITY CHECK

### 95%

**Of respondents are familiar with equipment lifecycle stages, but just over half have a proactive modernization strategy** [4]

***— ABB Global Lifecycle Management Report***

### $170,000 per hour (average)

**Reported cost of unplanned downtime across industries, with food and beverage often underestimating true impact**

***— ABB Global Research***

### 33%

**Manufacturers reporting no modernization activity in the past two years, despite rising disruption risk**

***— ABB Global Research***

## Final takeaway

Modernization succeeds when it happens before failure forces it — and when it is guided by clarity rather than urgency.

# TECHNICAL APPENDIX

Architectures, Lifecycle Maps & Engineering Reference Models

## Appendix A

### Automation Lifecycle Risk Map (What "Aging" Actually Looks Like)

Most plants think in terms of *old vs. new*. Engineers know the real distinction is **supported vs. supportable vs. constrained**.

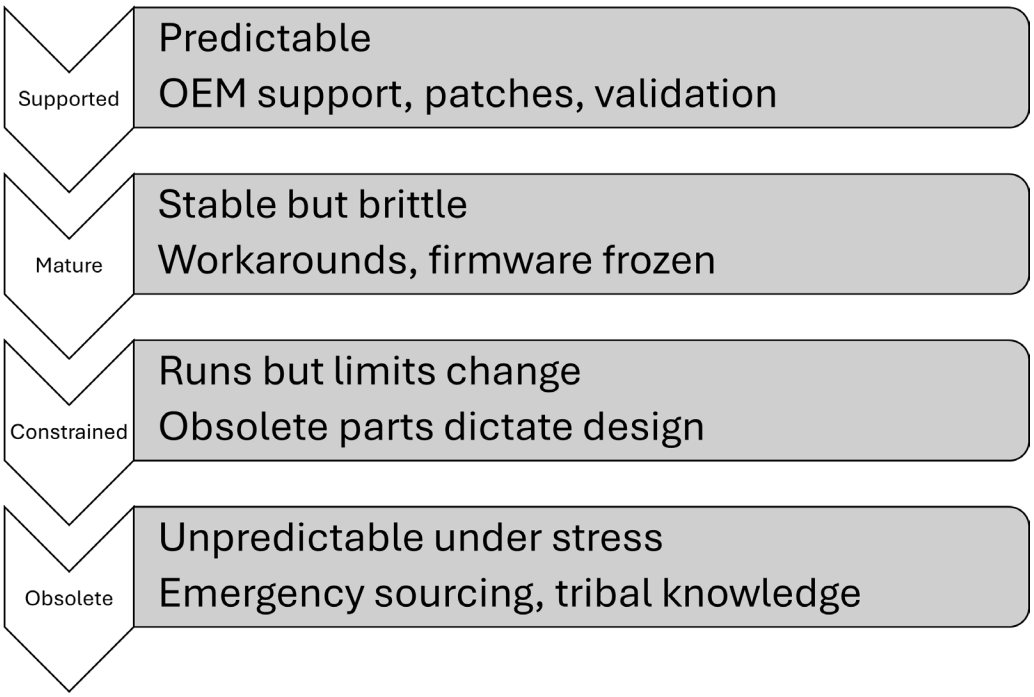
Below is a **practical lifecycle model** you can apply to any control system, subsystem, or line.

**Key engineering insight:**

**Risk accelerates fastest in the "constrained" stage**, not at end of life.

**This is where:**

- Firmware cannot be updated safely
- Cybersecurity controls cannot be validated
- Recovery paths become person-dependent



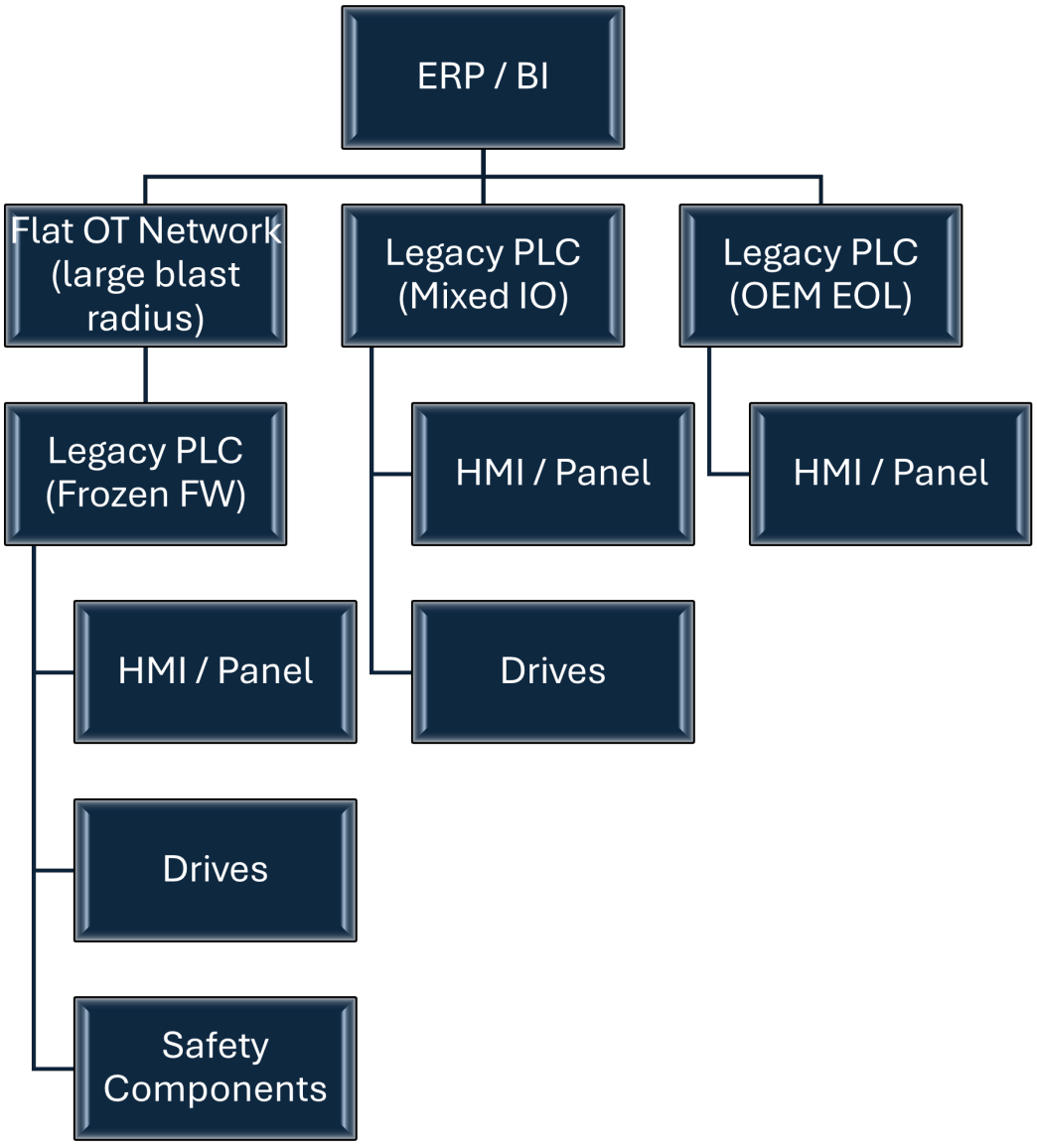
# Appendix B

## Brownfield Automation Architecture (Before Modernization)

This is the **most common architecture** we see in Food & Beverage plants.

### What engineers experience in this state

- Alarms without context
- Restart behavior varies by fault location
- Changes avoided because "it works today"
- Cyber or network issues indistinguishable from control faults



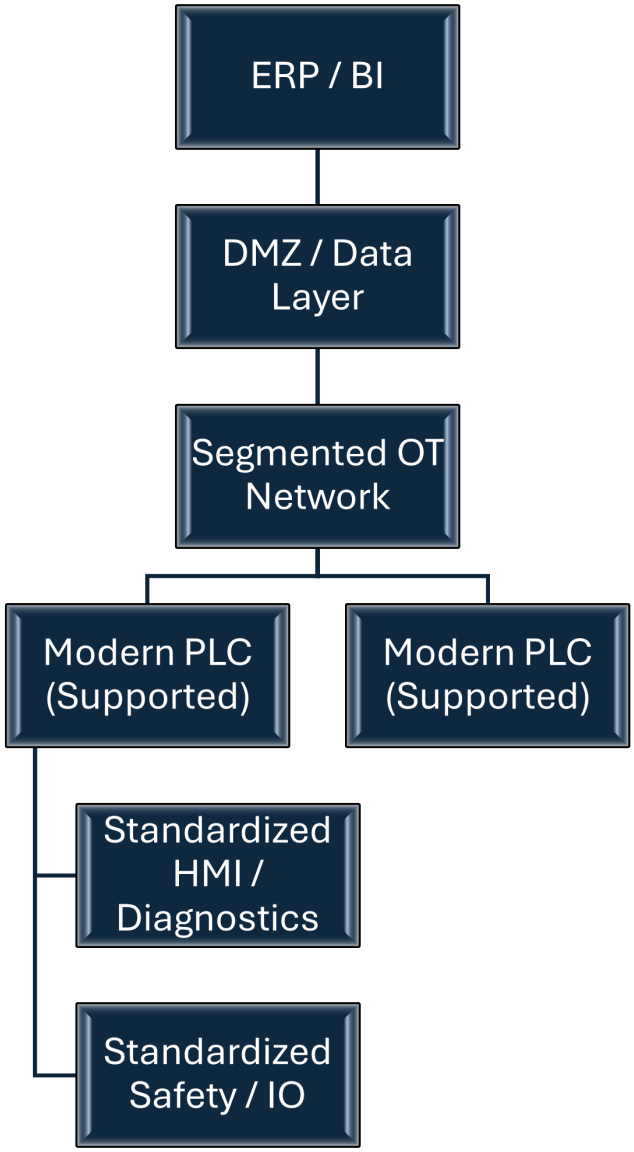
# Appendix C

## Modernized, Supportable Architecture (After)

This is not greenfield.
This is brownfield done intentionally.

### What changes operationally

- Faults are isolated by segment
- Restart logic is deterministic
- Cyber events don't look like "ghost faults"
- Engineers can reason about system behavior again



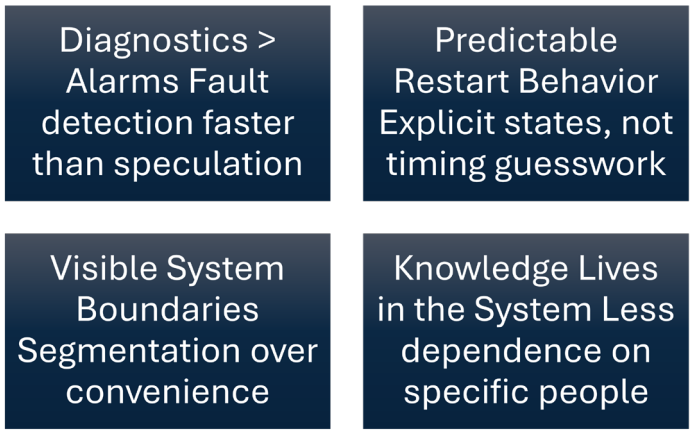
# Appendix D

## Recovery-Focused Design Principles (What Actually Reduces Downtime)

Most downtime cost is driven by **recovery uncertainty**, not failure frequency.

Design systems so that:

- **Fault detection is faster than fault speculation**
  - Diagnostics > Alarms
- **Restart behavior is predictable**
  - Explicit States, Not Implicit Timing
- **System boundaries are visible**
  - Segmentation Over Convenience
- **Knowledge lives in the system**
  - Less Dependence on Specific People



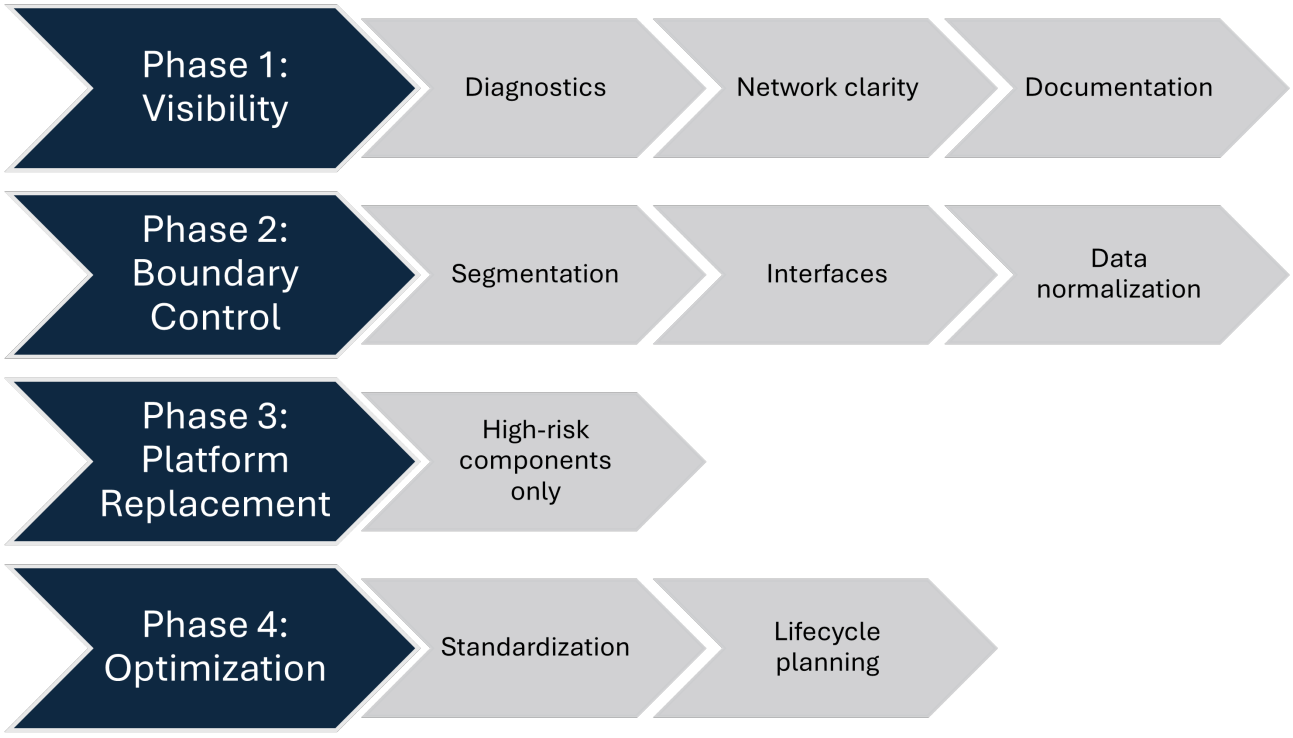
# Appendix E

## Modernization Without Shutdown — Engineering Pattern

This is the pattern used in the syrup room.

### Why this works

- Each phase reduces risk, not increases it
- Production continues throughout
- Failures don't compound across phases
- Engineers maintain confidence at every step



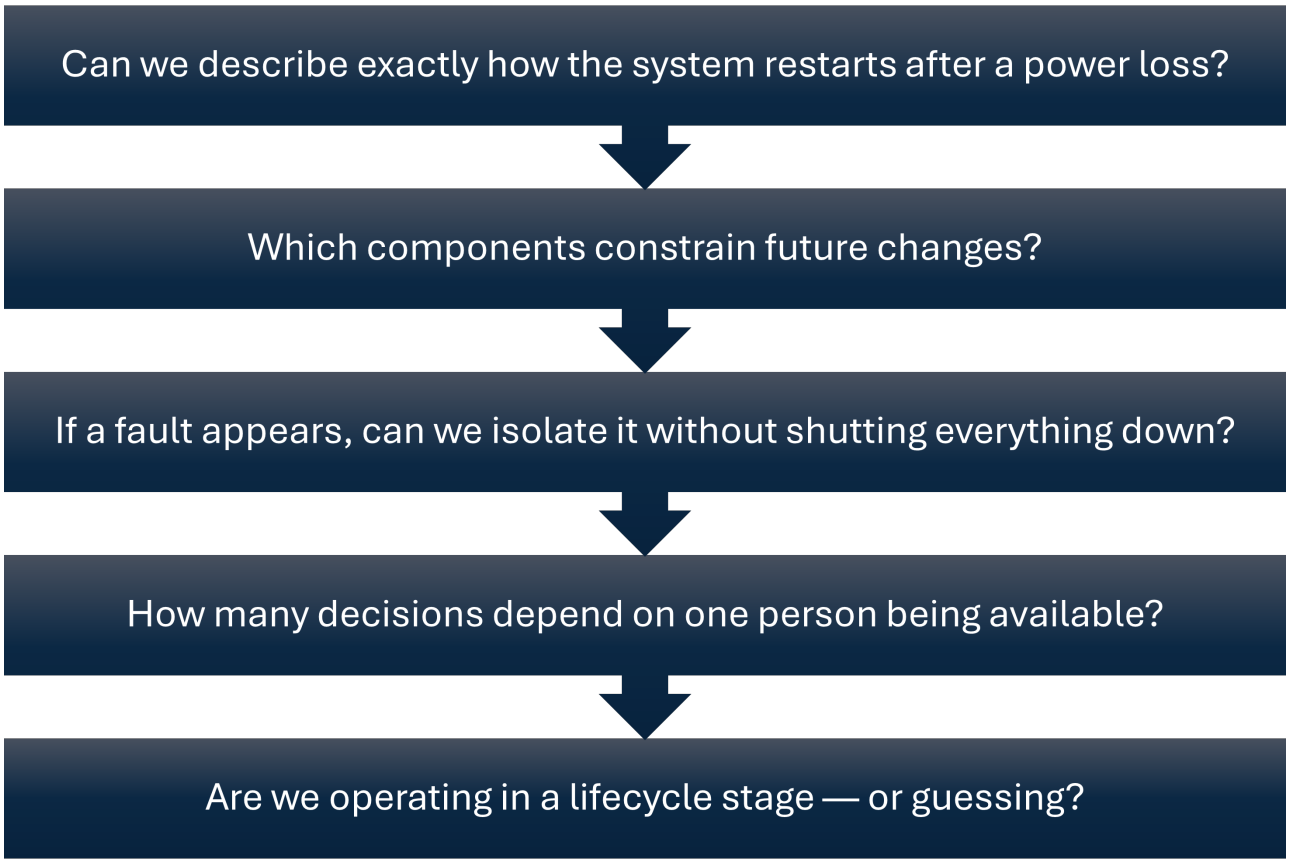
# Appendix F

## Practical Questions Engineers Can Ask Tomorrow

Use these to evaluate any system:

- Can we describe exactly how this system restarts after a power loss?
- Which components constrain future changes?
- If a fault appears, can we isolate it without shutting down everything?
- How many decisions depend on one person being available?
- Are we operating in a lifecycle stage — or guessing?

If those questions are uncomfortable, you're already in modernization territory.



# Appendix G

## Sources


1. ARC Advisory Group. *Managing the Lifecycle of Industrial Automation Assets.* Rockwell Automation White Paper. https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/gmsa-wp002_-en-p.pdf
2. Athena Controls. *Today's Challenges of Maintaining Legacy Control Systems.* https://www.athenacontrols.com/todays-challenges-of-maintaining-legacy-control-systems/
3. Rockwell Automation. *Product and Application Lifecycle Support. Citing ARC Advisory Group, 2010 Process Automation Study.* https://literature.rockwellautomation.com/idc/groups/literature/documents/sp/gmsg-sp010_-en-p.pdf
4. ABB. *Global Lifecycle Management Report: Modernization for Resilience.* https://www.abb.com/global/en/areas/motion/services/life-cycle-management/modernization-for-resilience-global-report
5. Manufacturing Digital (ABB Survey). *Unscheduled Downtime Costs and Operational Impact.* https://manufacturingdigital.com/procurement-and-supply-chain/unscheduled-downtime-costs-us-125-000-per-hour-abb-survey
6. Oxmaint. *Downtime Cost Impact on Food Manufacturing Operations.* https://oxmaint.com/industries/food-manufacturing/downtime-cost-impact-food-manufacturing
7. U.S. Food & Drug Administration. *FSMA Final Rule on Requirements for Additional Traceability Records for Certain Foods* https://www.fda.gov/food/food-safety-modernization-act-fsma/fsma-final-rule-requirements-additional-traceability-records-certain-foods
8. Reagan Udall Foundation for the FDA. *Industry Roundtable Series on the FSMA Final Rule on Requirements for Additional Traceability Records for Certain Foods* https://reaganudall.org/sites/default/files/2025-06/Food%20Traceability%20Top-Line%20Summary%20Final_0.pdf
9. Cognex. *Modernizing Brownfield Manufacturing Environments Without Disruption.* https://www.cognex.com/en/tools-and-resources/resource-center/modernizing-without-disruption
10. Xentara. *Brownfield Automation Retrofit Case Study.* https://kb.xentara.io/articles/case-study-brownfield-retrofit-and-upgrade
11. Food Logistics / Plex. *Building a Traceable Food Chain: A Comprehensive Guide to FSMA Compliance.* https://www.foodlogistics.com/safety-security/food-safety/article/22921687/plex-building-a-traceable-food-chain-a-comprehensive-guide-to-fsma-compliance
12. ServiceMax / Vanson Bourne. *After the Fall: The Costs, Causes and Consequences of Unplanned Downtime.* https://www.panelbuilderus.com/wp-content/uploads/2020/11/After-The-Fall-whitepaper-updated-global-numbers-FINAL-refresh.pdf
13. Dematic. *The Rise of Brownfield Automation: Reinvesting in Existing Infrastructure for Supply Chain Success.* https://www.dematic.com/en-us/insights/articles/brownfield-automation-reinvesting-in-existing-infrastructure-for-supply-chain-success/
14. Cybersecurity and Infrastructure Security Agency (CISA). *Foundations for OT Cybersecurity: Asset Inventory Guidance for Owners and Operators.* https://www.cisa.gov/resources-tools/resources/foundations-ot-cybersecurity-asset-inventory-guidance-owners-and-operators


*Portions of this ebook were drafted and edited with the assistance of an artificial intelligence tool (Copilot). All content was reviewed, revised, and finalized by the author.*

STANDARD
ELECTRIC SUPPLY CO.

# YOUR PARTNER IN SUCCESS

Standard Electric Supply Co. is a full-line electrical distributor, emphasizing industrial automation and control products with a focus on the OEM and MRO marketplace. Established in 1919, we are a fourth-generation, family-owned company committed to supplying our customers with the very best products and solutions. As a family-owned business, we have the ability to tailor specific solutions and adapt quickly to customer needs. This is our advantage over other suppliers. After more than 100 years in business, we know what it takes to make a partnership succeed.

- Inventory Management Solutions
- Enclosure Modification Solutions
- Custom Assemblies & Kitting
- Technical Solutions Team
- Training & Seminars
- Repair Services
- National Customer Solutions
- Value Engineering Assessment


EB-01 | Rev A | MAR 2026

800-776-8222 WISCONSIN | ILLINOIS | INDIANA www.standardelectricsupply.com