

Why most manufacturing software implementations fail, and how to prevent it.

Five reasons rollouts struggle, two real-world implementations, and nine practices that turn new software into a lasting operational advantage.



Adoption is decided on the floor, not in the demo.

Jason C. Dixon

Solutions Engineer · Modular & prefab manufacturing
Hamlet, North Carolina · Available for travel across North America

jasoncdixon.com

jason.c.dixon@gmail.com
[linkedin.com/in/jasoncdixon](https://www.linkedin.com/in/jasoncdixon)

August 2026

CONTENTS

What's inside

-
- 01 The wrong people are leading the implementation
 - 02 Every factory needs a subject matter expert
 - 03 Poor planning creates long and expensive implementations
 - 04 Don't overlook the hardware and IT infrastructure
 - 05 Companies stop using the software as a management tool
 - 06 Real-world example: fast doesn't always mean successful
 - 07 Real-world example: when culture becomes the bottleneck
 - 08 Nine best practices for a successful implementation
 - 09 Final thoughts and key takeaways
-

Technology isn't usually the problem

Over the past several years, I've helped manufacturing companies implement software ranging from ERP and MES systems to production tracking, quality management, inventory control, and labor management platforms.

I've seen companies completely transform their operations in a matter of weeks. I've also seen companies spend thousands of dollars on software only to abandon it months later.

The interesting part is that the software usually wasn't the problem. Most manufacturing software implementations fail because companies underestimate the operational changes required to make the software successful.

Software simply exposes the strengths and weaknesses that already exist inside an organization.

If your processes are disciplined, software accelerates your success. If your processes lack accountability, software simply makes those problems more visible.

After working with manufacturers across the modular, panelized, and prefab industries, I've found five common reasons implementations struggle.

REASON 01

The wrong people are leading the implementation

One of the biggest mistakes I see is assigning software implementation to the people who simply use the software every day. Those employees absolutely need to be involved; their input is critical because they're the ones performing the work. But they shouldn't own the implementation.

Successful implementations require someone whose job is making the project successful, not someone trying to squeeze implementation work between their normal daily responsibilities. The implementation leader should have enough authority to make decisions, remove roadblocks, and hold people accountable. Just as importantly, they need to genuinely believe in the project.

When leadership assigns software to someone who has little ownership, little authority, or little interest, the project almost always loses momentum.

REASON 02

Every factory needs a subject matter expert

Another major reason implementations struggle is the absence of a dedicated Subject Matter Expert. An SME understands three things:

- How the factory actually operates.
- How the software is designed to work.
- How to bridge the gap between those two worlds.

Without someone filling this role, companies often configure software based on assumptions rather than actual workflows. The result is frustration from employees who feel like the software "doesn't fit."

In many situations, hiring an implementation manager, working with a consultant, or bringing in a trainer from the software company is one of the best investments a manufacturer can make. The upfront cost is usually far less than the cost of months of delays, poor adoption, and bad data.

REASON 03

Poor planning creates long and expensive implementations

Many organizations focus almost entirely on one question: *"How much does the software cost?"* The better question is: *"What will it cost us to successfully implement the software?"*

That includes much more than the purchase price. A proper implementation plan should consider:

- Software licensing
- Hardware requirements
- Implementation labor
- Employee training
- Process changes
- Downtime during rollout
- Internal project management
- Ongoing support

Perhaps the biggest planning mistake is unrealistic expectations. Factories often assume implementation will happen quickly. In reality, implementation takes as long as the

organization takes to develop disciplined habits.

| *The software isn't slow. Changing habits is.*

I've often joked with clients that implementation usually takes about twice as long as they expect.

REASON 04

Don't overlook the hardware and IT infrastructure

One of the most overlooked aspects of a successful software implementation has nothing to do with the software itself: it's the hardware and network infrastructure that support it. I've seen companies invest significant time and money into selecting the right software only to discover, during implementation, that their factory isn't ready to support it.

Before going live, ask yourself questions like:

- Do we have reliable Wi-Fi coverage throughout the entire facility?
- Are there dead zones on the production floor where tablets or scanners won't stay connected?
- Do we have enough tablets, computers, barcode scanners, printers, or other hardware for employees to perform their work efficiently?
- Is our network secure and capable of handling additional connected devices?
- Who will support the hardware if something fails during production?

These questions should be answered before implementation begins, not after employees are already trying to use the system. For many manufacturers, partnering with an experienced IT company or assigning dedicated internal IT personnel is a worthwhile investment. They can ensure wireless access points are properly positioned, devices are configured correctly, security policies are in place, and hardware is ready before the software goes live.

Nothing frustrates employees more than being asked to use new software on unreliable equipment or over an unstable network. If the tablet continually loses its connection or the barcode scanner won't communicate with the system, employees quickly lose confidence, not only in the hardware but in the software itself.

A successful implementation depends on the entire technology ecosystem working together. The software may be the centerpiece, but reliable hardware, dependable Wi-Fi, properly configured devices, and responsive IT support are all critical components of long-term success.

REASON 05

Companies stop using the software as a management tool

Many factories successfully go live. Unfortunately, that's where many implementations begin to fail. People continue entering information. Reports continue running. But no one actually uses the data, and the software becomes nothing more than a digital checkbox.

Instead of asking:

- What are today's production bottlenecks?
- Which department is falling behind?
- Where is rework increasing?
- Which KPIs are changing?

Leadership simply collects data because "that's what we're supposed to do."

Software should drive conversations. It should improve decisions. It should expose problems before they become expensive. Without accountability, even great software eventually becomes ignored.

Real-world example: fast doesn't always mean successful

One of the fastest implementations I've ever completed was at a panelized manufacturing facility. We successfully launched the software in just two weeks. From a technical perspective, everything worked, and the project manager began using reports generated by the software to monitor weekly production.

There was just one problem. Employees weren't entering accurate information. As a result, the reports looked impressive, but they weren't telling the truth.

Instead of auditing the data and coaching the team toward better habits, the organization slowly returned to its old paper-based processes. Ironically, they ended up relying on paper again while still struggling with inaccurate information, because the real issue had never been addressed.

| *The software wasn't the failure. Accountability was.*

Real-world example: when culture becomes the bottleneck

Another manufacturer struggled with implementation for months. Employees would use the software for a few weeks, stop using it, then try again. The owner became frustrated because the reports couldn't be trusted.

I was asked to walk through the factory and identify the root cause. After spending time observing operations, interviewing leadership, and watching daily workflows, it became clear the software wasn't the biggest issue. Leadership consistency was.

Employees naturally follow the standards demonstrated by leadership. If management treats software as optional, employees will too.

After presenting my findings, leadership made several organizational changes. Later, I returned to conduct leadership training focused on accountability, communication, and operational management. Once leadership changed its approach, software adoption dramatically improved. Culture had been slowing progress far more than technology ever was.

Best practices for a successful implementation

1. Appoint a dedicated implementation leader

Give one person ownership of the project. Ideally, this person also serves as the Subject Matter Expert and has a backup who understands the system and can keep the project moving when they're unavailable.

2. Evaluate software before buying it

Create evaluation criteria before speaking with vendors. Consider the problems you're trying to solve, budget, required functionality, ease of use, implementation timeline, integration requirements, and long-term scalability. Remember that implementation time is often more expensive than the software itself.

3. Limit the evaluation period

Software evaluations shouldn't last forever. Most organizations can evaluate multiple platforms within **30 to 45 days**. Long evaluation periods usually delay improvement rather than improve decisions.

4. Demand proof of concept

Don't simply believe marketing demonstrations. Request a trial. Use your own data. Test real workflows. Ask vendors to prove they can solve your specific operational problems. If possible, ask to speak with an existing customer who has implemented the software successfully. An experienced software consultant from the vendor who understands your business can also provide tremendous value during the evaluation process.

5. Build a detailed implementation roadmap

Every implementation should include milestone dates for hardware installation, network readiness, software configuration, user training, pilot testing, go-live, reporting, and performance reviews. Everyone should understand what success looks like before implementation begins.

6. Review progress frequently

Meet regularly throughout implementation. If milestones begin slipping, determine why immediately. Document delays. Create recovery plans. Accountability should exist at every level of the organization.

7. Support your implementation team

One of the strongest indicators of success is visible executive support. When ownership consistently reinforces the importance of the project, employees take it seriously. Celebrate milestones, recognize progress, and show employees the value the software is creating.

8. Conduct a 90-day review

Going live isn't the finish line. After three months, evaluate software usage, data quality, reporting accuracy, user adoption, process compliance, and opportunities for improvement. Many companies discover they're only using a fraction of what their software can actually do.

9. Perform an annual operational assessment

One year after implementation, bring in someone from outside your organization. An experienced consultant can identify untapped software capabilities, inefficient workflows, cultural barriers, reporting gaps, and missed automation opportunities. Fresh eyes frequently uncover opportunities internal teams no longer notice.

Final thoughts

Manufacturing software doesn't improve factories. People improve factories. Software simply gives those people better information, greater visibility, and stronger tools for making decisions.

Successful implementations happen when organizations invest just as much effort into people, leadership, planning, infrastructure, and accountability as they do into selecting the software itself.

Technology is an investment. Operational discipline is what produces the return on that investment.

The manufacturers who consistently succeed aren't the ones with the most expensive software. They're the ones who build the right culture around it.

Key takeaways

A successful manufacturing software implementation is far more than installing a new system; it's leading organizational change. Keep these principles in mind:

- **Evaluate** software based on your operational needs, not just features.
- **Test** the software using real-world workflows before purchasing.
- **Plan** for the total cost, including hardware, training, IT infrastructure, and implementation resources.
- **Assign** a dedicated implementation manager or SME to lead the project.
- **Prepare** your factory's technology by ensuring Wi-Fi, tablets, scanners, computers, and printers are ready before go-live.
- **Create** clear implementation milestones and hold regular progress reviews.
- **Hold everyone accountable**, from leadership to operators, for following the new processes.
- **Audit** software usage after go-live to ensure accurate data and proper adoption.
- **Continuously improve** by reviewing reports, KPIs, and user feedback.
- **Bring in outside expertise** when needed. A trusted consultant can identify operational gaps, cultural challenges, and untapped capabilities internal teams may overlook.

Software alone will never transform a factory. The combination of **the right people, the right planning, the right technology, and consistent accountability** is what turns a software

implementation into a lasting competitive advantage.

ABOUT THE AUTHOR

Planning a rollout, or rescuing one?

Jason C. Dixon is a Solutions Engineer working with modular, panelized, and prefab manufacturers across North America: operational assessment, quality systems, and software implementation. He has built quality systems from the inside as a QC manager and implemented operational technology from the outside, so the advice above comes from both sides of the rollout.

jasoncdixon.com · jason.c.dixon@gmail.com · linkedin.com/in/jasoncdixon

Hamlet, North Carolina · Available for travel across North America

© 2026 Jason C. Dixon. Shared for the benefit of manufacturers evaluating or recovering an implementation.