GoBlock®
Modernize Your Warehouse

Did I Get What I Paid For?

How to Recover the ROI You Expected from Warehouse Automation

Many companies invest heavily in warehouse automation only to find performance and ROI fall short of expectations. This white paper examines common causes, ranging from poor data and design rigidity to change orders, consultant bias, and flawed growth forecasts, and offers strategies to identify issues, remediate underperformance, and design flexible systems that deliver lasting value.



© 2026 GoBlock USA LLC
All rights reserved.



Part I: Is this You?

You are a warehouse/distribution leader at a growing company that has just spent several years implementing a major automation investment

The project required millions of dollars in capital, significant organizational focus, and countless decisions about how your operation would run in the future. The design promised higher throughput, lower labor dependency, and a more scalable operation built to support growth.

During the design phase, everything made sense. The models showed how products would flow. Labor calculations demonstrated efficiency gains. The system appeared balanced, deliberate, and capable of meeting peak demand. You approved the investment, believing it would fundamentally improve how your warehouse operated.

Then reality set in.

As implementation progressed, compromises were made. Some were driven by schedule pressure. Others came from equipment availability, layout constraints, or late operational feedback. Each compromise felt manageable in the moment. The priority was getting the system installed and live, often ahead of an upcoming peak season.

When the system went live, it worked, but not quite as expected. Throughput was more constrained than modeled. Labor reductions did not fully materialize. Downtime lingered longer than planned. To keep orders moving, you capped customer demand, added temporary labor, or introduced manual workarounds that were never part of the original design.

In the best cases, you made it through peak with extra effort, additional labor, and a lingering sense that the system was not delivering what it should.

In the worst cases, performance suffered enough that service levels declined, costs increased, and the automation investment became a topic of concern at the executive level. Questions surfaced internally (and in some cases externally) about *"why was such a significant investment not producing the expected return?"*

As you inquired, you may have heard assurances that the issues would be resolved by the next peak, and the system would perform as expected. Perhaps you had heard that before.

But you are still left asking a question that is both simple and difficult to answer:

"Did I get what I paid for?"

This white paper is written for leaders in that position. It does not assume that every automation project fails, nor that every issue has the same cause. Instead, it reflects what we have seen across warehouse and distribution environments as systems go live, businesses evolve, and operational realities collide with original design assumptions.

In Part II, we present a series of real-world stories that illustrate common ways automation projects can get off the rails. Some of these stories may sound uncomfortably familiar. Others may not apply to your situation at all. They are not meant to assign blame, but to highlight patterns that affect performance, labor, and return on investment.

For the stories that resonate, we offer practical next steps. These are not redesign prescriptions or vendor defenses. They are structured ways to step back, objectively assess what has changed, and determine what can realistically be improved within the system you already own.

In Part III, we outline a disciplined approach to doing exactly that: *improve what you already have.*

GoBlock®
Modernize Your Warehouse

Table of Contents

Did I Get What I Paid For? 1
Part I: Is this You? 3
Part II: How Good Projects Go Off Track 6
Chapter A. “I Have a Day Job” 7
Chapter B. Death by a Thousand Cuts 9
Chapter C. “Don’t Worry, It Will Work Out” 11
Chapter D. Incomplete Inputs and Unpredictable Outcomes 13
Chapter E. The Business Changed 15
Chapter F. When Schedule and Cost Win 17
Chapter G. The Sensitivity of Growth 19
Part III: The Disciplined Way Forward 21
Phase A: Understanding the Base Design 24
1. Business Case and ROI Intent 24
2. Design Assumptions and System Intent 25
3. Technology Performance Specifications 26
4. Commercial Agreements and SLAs 26
Phase B: Current State Study and Analysis 27
5. Business Reality and Demand Profile 27
6. Engineering Process Study and Bottleneck Identification 28
7. KPI Visibility and Performance Control 28
8. Downtime and Stability Analysis 29
9. Labor Model Validation 29
Phase C: Findings and Recommendations 30
10. Flow Conformance and Workarounds 30
11. Financial Impact: Unit Cost, Downtime Cost, and Quality Cost 31
12. Bottleneck Expansion and Improvement Options 31
13. Support Model Effectiveness 32
Engagement Deliverables and Outcomes 33
Engagement Timeline and Responsibilities 34

Part II:

How Good Projects Go Off Track

Chapter A.
“I Have a Day Job”

When Leadership Bandwidth Runs Out

You had just survived another peak. Volumes ran hot, downtime cropped up at the worst moments, and you leaned on temporary labor to keep orders moving. It was exhausting, but the operation held together because you and a few other leaders stepped in constantly to keep things from breaking.

You knew you needed to increase your warehouse capabilities, and leadership agreed with you.

Shortly after peak, leadership announced that a consulting firm would be coming in to study the operation and design those new capabilities. You were told you would provide the “operations perspective” in that design effort.

At first, it felt like recognition! You knew the operation better than anyone. But reality set in quickly. Orders still needed to ship every day. Issues from peak still needed root cause fixes. Meanwhile, you were expected to attend design workshops, review process flows, validate labor assumptions, and think through how future-state operations would actually run.

You showed up as best you could. You reviewed documents late at night. You trusted that processes that looked good on paper would work in practice. When something did not quite feel right, you told yourself you would deal with it later, after go-live, when things slowed down. The project had momentum and deadlines, and you did not want to be the one who slowed it down.

As the months went on, continuity started to slip. The Project Management Office changed. Some workstream leaders rotated. Conversations between Ops and IT lost detail. Assumptions made early were no longer revisited. The design was locked in, even though the people who were supposed to run it had not truly understood the new processes yet.



Then, after go-live, the gap became obvious. The system technically worked, but only if you and your team stayed deeply involved every day. Labor and throughput targets were missed unless managers intervened constantly. Informal workarounds filled the gaps. When key leaders were out, performance dipped. The automation had not failed, but it required a level of sustained leadership attention that was never realistic.

If this happened to your automation project, what can you do about it now?

The next step is to objectively evaluate whether the automation is being asked to operate under leadership, ownership, and operating assumptions that are no longer realistic. By revisiting what human supervision the system was designed to require, measuring how management involvement, decision-making, and performance control actually occur today, and identifying where gaps in ownership or operating rhythm are constraining results, organizations can reset accountability and stabilize performance without relying on informal workarounds or overextended leaders.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Step 2), and Phase B: Current State Study and Analysis (Steps 6, 7, and 9), and Phase C: Findings and Recommendations (Step 10).

Chapter B.
Death by a Thousand Cuts

Project Change Orders

When you approved the automation design, it felt thoughtfully balanced. Bottlenecks were understood. Labor was the constraint in certain areas, but that was OK. Conveyance, sortation, and induction were sized to support that balance, and the ROI model assumed those relationships would hold.

Then, change orders started showing up. One came from operations when someone realized a process felt too tight and required adjustment. Another came from the equipment vendor when equipment availability changed. Others were driven by layout constraints, safety feedback, or schedule pressure. Each change was small. Each one made sense in isolation.

One change involved the design of an induction workstation. The original design included lift tables that allowed pallets to be raised to an ergonomic height, increasing induction rates. Later, a safety review recommended removing those tables to widen the drive aisle. It looked like a savings and reduced the capital cost. But no one recalculated what that meant for the induction rate.

Then, after go-live, you felt it. Induction took longer. Picking windows shortened. To compensate, you either added labor or accepted lower throughput. Multiply that by a dozen similar changes, and the system no longer resembled the one the business case was built on.

Looking back, the issue was not poor decision-making. It was that no one stepped back to assess what the *sum* of those changes did to the system as a whole.

The ROI quietly bled out, one paper cut at a time.



If this happened to your automation project, what can you do about it now?

The next step is to objectively evaluate how post-design change orders, taken individually and in combination, have altered system performance, labor requirements, and return expectations relative to the original design intent. By revisiting what the system was designed to deliver, measuring how throughput, induction rates, and operating costs are performing today, and identifying which changes are materially constraining capacity or eroding value, organizations can separate acceptable compromises from those that are driving underperformance and take focused action to recover lost ROI.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Steps 1, 2, and 4) and Phase C: Findings and Recommendations (Steps 11 and 12).

Chapter C. “Don’t Worry, It Will Work Out”

Optimism That Kept the Project Moving

You completed a detailed design with your consultant and felt confident moving into procurement for equipment and systems. The team said “Go!”

Then, reality intervened. Lead times stretched. Certain equipment was unavailable. To protect the schedule, substitutions were proposed. You were told the alternatives were equivalent.

It felt like buying a new sports car that you could not test drive. You picked the color and the features based on specs and reviews. But months later, you were told the color was unavailable. Then, certain parts would be swapped from a different model. Then servicing would be split across vendors. Each change came with reassurance that you would get what you paid for.

After go-live, the system technically met specs, but it felt different. Downtime recovery took longer. Maintenance required more coordination. Support felt fragmented. You realized you were operating a system that no longer matched what you thought you bought.

If this happened to your automation project, what can you do about it now?



The next step is to objectively assess where substitutions and design changes made during procurement and implementation altered performance, reliability, or serviceability relative to what was originally specified. By revisiting what was agreed to, measuring how the system actually performs today, and separating optimistic explanations from operational reality, organizations can focus on practical improvements rather than continuing to manage around unintended complexity.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Steps 2, 3, and 4) and Phase B: Current State Study and Analysis (Steps 6 and 8).

Chapter D. Incomplete Inputs and Unpredictable Outcomes

Design Data

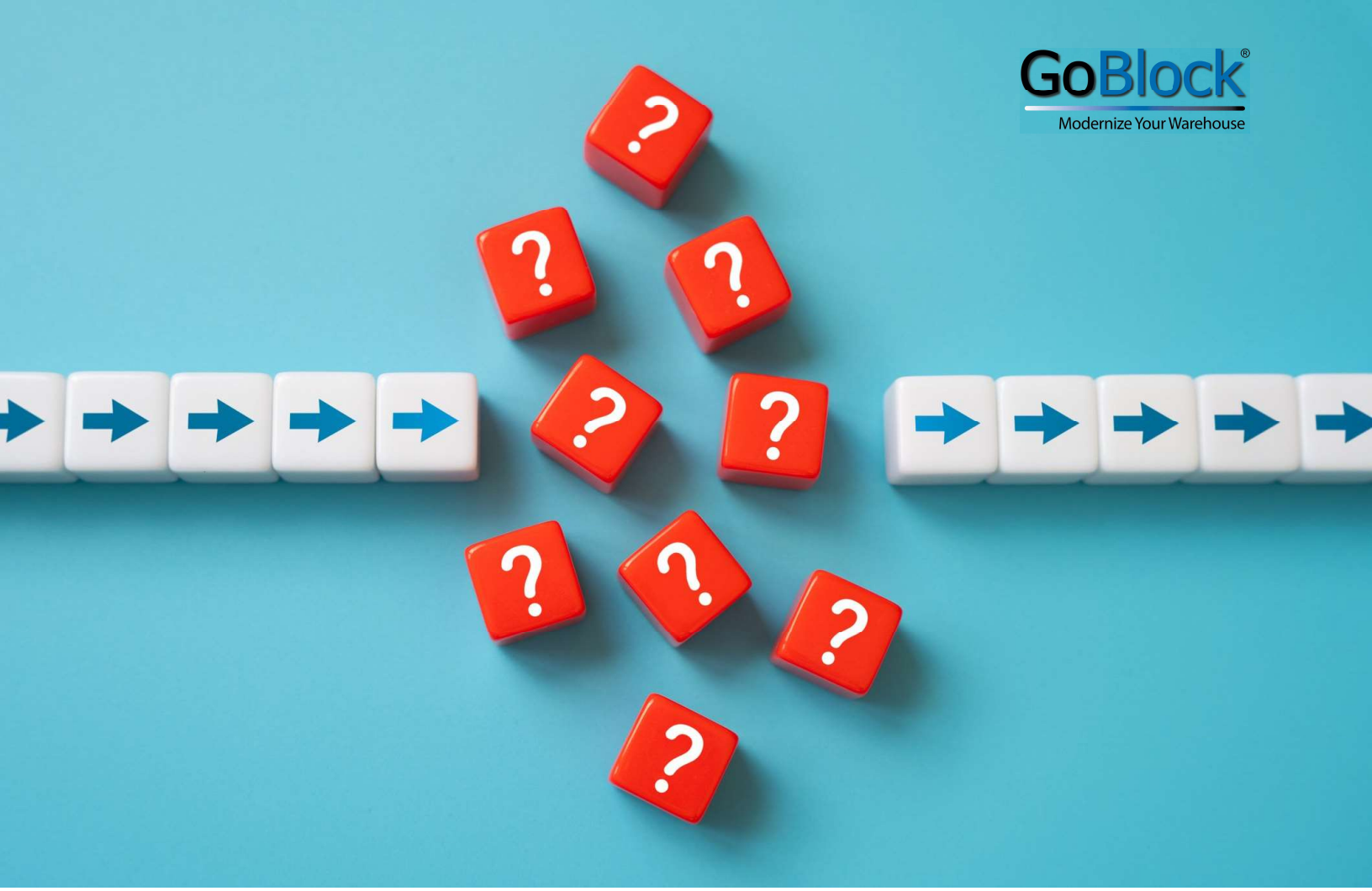
When leadership asked you to explore automation, you were asked to provide all the operational information required to design the system. You pulled what historical data was available, or that you could collect in the time allowed. When they requested two years of line-level data, inventory snapshots, receiving history, and SKU master details, it quickly became clear that gathering all of it would take more time than you had. Order counts were summarized at a high level. SKU data was incomplete. Labor reporting existed, but not at the level of detail the consultants asked for.

You felt pressure to move forward anyway. Leadership wanted answers. Consultants were ready to size the technology. So, assumptions filled the gaps. Averages replaced variability. You told yourself that the operation was fairly stable and that the system would absorb the differences.

And, during design, everything penciled out great. Throughput targets were achievable. Labor models looked efficient. The business case was approved. Implementation moved forward.

Then, after go-live, the cracks appeared. Peak behavior did not resemble the averages used in the design data. Certain SKUs behaved very differently from what was expected. Labor was required in ways that had not been modeled. What looked like a minor data shortcut early in the project turned into chronic operational friction.

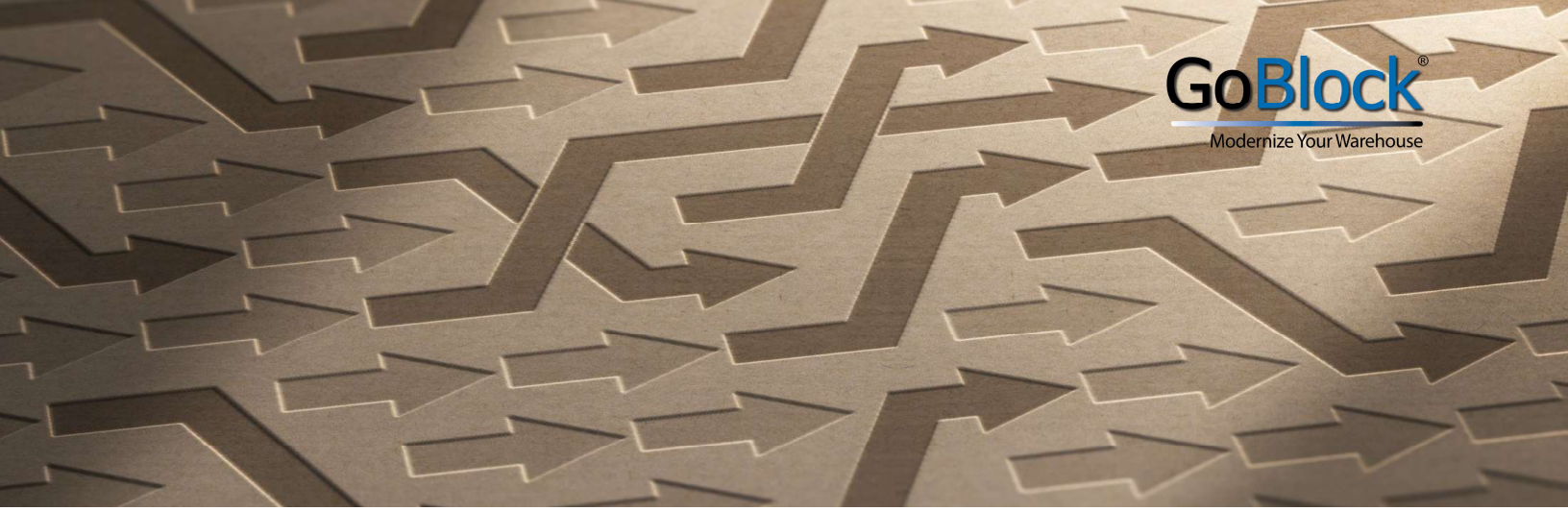
You found yourself compensating with additional labor, manual intervention, and revised operating rules. The system did what it was designed to do, but not what the business actually needed. In hindsight, the issue was not technology selection. It was that the design reflected the data that was easiest to provide, not the data that truly described how your operation behaved.



If this happened to your automation project, what can you do about it now?

The next step is to validate whether the data and assumptions used to design the system reflect how the operation actually behaves today. By comparing original modeling inputs to today's real volumes, SKU behavior, labor flows, and variability, organizations can replace those inherited assumptions with measured reality and make informed decisions about how to improve performance.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Steps 1 and 2) and Phase B: Current State Study and Analysis (Steps 5 and 6).



Chapter E.
The Business Changed

The System Didn't

When the automation first went live, it felt like a win. Walking and transport were reduced. Productivity improved. The operation stabilized. The system fit the business you were running at the time.

Then things changed. New product lines launched. An acquisition was consolidated into the same facility. Suddenly, SKUs with different dimensions, weights, and handling requirements entered the flow. Some of your highest-volume products no longer fit within the automated system that had been designed around a different profile.

You adapted as you could. Certain SKUs bypassed automation. Manual processes were layered back in. Special handling rules were created. At first, these were temporary exceptions. Over time, they became permanent.

Throughput slowly declined. Labor crept back into areas that were supposed to be automated. The system still ran, but it was no longer doing the heavy lifting it once did. When you looked at performance metrics, it was hard to tell whether the issue was execution or design fit.

In reality, the automation had not failed. The business it was supporting had simply evolved beyond the assumptions embedded in the original design.

If this happened to your automation project, what can you do about it now?

The next step is to assess how today's product mix and demand patterns interact with the system's intended capabilities and constraints. By measuring where current products and order profiles are creating friction or bypasses, organizations can determine whether performance issues stem from execution or from a loss of design fit, and identify practical options to restore flexibility.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Step 2), and Phase B: Current State Study and Analysis (Steps 5 and 6), and Phase C: Findings and Recommendations (Step 12).



Chapter F. When Schedule and Cost Win

But Flexibility Pays

Early in the project, “flexibility” was identified as a key requirement, and multiple design options were considered to provide that flexibility.

One decision involved a sorter design. You had two options: implement a single loop that handled both inbound and outbound flow, or split the functions across two simpler sorters. The single-loop met throughput targets and it was more flexible, but required more complex controls and higher capital. The split sorter design would cost less and be easier to implement.

The team chose to split the sorters. At the time, it felt like the right decision.

As the project progressed, additional changes were introduced. Peer feedback prompted second-guessing. Late-stage modifications affected engineering, procurement, and controls. Each change added complexity and tightened the schedule. Flexibility was no longer key. The focus was getting it done and live.

Then, after go-live, the system worked, but recovery from issues was slow. A failure in one area had an outsized impact. Those outsized impacts required experienced managers to solve. The system was efficient when everything went right, and unforgiving when it did not.

Looking back, the problem was not the design choice itself. It was that rigidity and resilience had been traded without fully understanding how the system would behave under stress.



If this happened to your automation project, what can you do about it now?

The next step is to objectively evaluate where design rigidity limits resilience and increases sensitivity to disruption. By comparing intended flexibility with actual recovery behavior and operational burden, organizations can distinguish between acceptable tradeoffs and constraints that undermine long-term stability.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Steps 2, 3, and 4) and Phase C: Findings and Recommendations (Steps 12 and 13).

Chapter G.
The Sensitivity of Growth

The Assumption That Breaks Payback

When you justified the automation investment, the growth assumptions felt reasonable. The forecast was based on recent trends. Layered on top were the expected new sales that you all anticipated to see as a business. And using that growth curve, the ROI model showed a strong payback. Leadership approved the spend.

Years later, reality diverged. In some areas, growth outpaced expectations, and the system became constrained sooner than planned. In others, growth slowed, and capacity sat idle. Either way, the original business case no longer described what was happening on the floor.

You found yourself explaining why a system that technically worked was not delivering the returns everyone expected. The issue was not execution. It was that ROI had been treated as a static calculation instead of something that needed to evolve with the business.

In hindsight, you realized that the most effective projects were the ones where growth scenarios had been modeled explicitly and where the growth sensitivity impacts were well understood. Best case, worst case, and everything in between. Systems that could be expanded modularly or phased over time based on this sensitivity aged far better than those built on a static assumption.



If this happened to your automation project, what can you do about it now?

The next step is to revisit the growth assumptions that shaped the original business case and compare them to how demand and complexity have actually evolved. By quantifying the financial impact of those shifts, organizations can reframe ROI as a living model and make informed decisions about whether to optimize, expand, or defer additional investment.

To learn more, reference the GoBlock methodology outlined in Part 3, which addresses this directly through Phase A: Understanding the Base Design (Step 1) and Phase C: Findings and Recommendations (Step 11).

Part III:

The Disciplined Way Forward

The “Did I Get What I Paid For?” engagement is a structured post-implementation assessment designed to determine whether a warehouse technology investment is delivering the performance and financial returns it was intended to provide.

The engagement focuses on fact-finding, root-cause identification, and actionable remediation. Its purpose is not system replacement, blame, liability, or vendor defense.

The service is delivered through a *three-phase assessment framework* designed to establish clarity, isolate root causes, and define practical paths forward. Each phase builds on the prior one and answers a specific executive question.

- **Phase A: What was designed and agreed?**
- **Phase B: What is the current state today?**
- **Phase C: How different is design from reality, and what can be done about it?**

This structure ensures that findings are grounded in documented intent, measured against real performance, and translated into actionable recommendations.


GoBlock®
Modernize Your Warehouse

Phase A: Understanding the Base Design

What was designed and agreed?

The assessment begins by establishing a clear, factual baseline of what the investment was intended to deliver. This phase removes ambiguity by anchoring the analysis to documented commitments rather than institutional memory.

1. Business Case and ROI Intent

We review the original business case to understand:

- How the investment was expected to generate a return.
- The assumptions around volume, labor reduction, service improvement, and cost savings.
- Payback period, capital phasing, and sensitivity to growth or variability.

This clarifies what “success” meant at the time of approval and which assumptions were most critical to ROI.



2. Design Assumptions and System Intent

We examine the operational and engineering assumptions embedded in the design, including:

- Process flows and operating modes.
- Throughput targets and utilization assumptions.
- Storage profiles and chosen media.
- Inbound and Outbound order profiles and variability.
- Labor models and staffing profiles.
- Expected operating behaviors and constraints.

This step identifies where simplifications, averages, or best-case assumptions were used to size the system.

3. Technology Performance Specifications

We review the performance specifications used to design and procure the technology, including:

- Rated capacities and performance envelopes.
- Availability, uptime, and recovery assumptions.
- Software, controls, and integration expectations.

This establishes what the technology was supposed to be capable of delivering under defined conditions.

4. Commercial Agreements and SLAs

Finally, we review contracts and service agreements with technology providers to understand:

- Performance guarantees and service level agreements.
- Support models, response times, and escalation paths.
- Responsibilities across equipment and systems vendors, IT, and operations.
- Change Orders after the initial contract

This defines what was contractually agreed upon versus what was assumed or implied.



Phase B: Current State Study and Analysis

What is the current state today?

With the design baseline established, the assessment shifts to understanding how the system actually performs in today's operation.

5. Business Reality and Demand Profile

We analyze how the business has evolved since its design, including:

- Actual volumes, peaks, and seasonality.
- Inbound and outbound order profiles.
- SKU count, velocity shifts, and physical characteristics.
- New customers, channels, or service commitments.

This determines whether the system is being asked to operate outside its original design envelope.

6. Engineering Process Study and Bottleneck Identification

An engineer-led process study is conducted to map real flows through the operation:

- Breakdown of discrete process steps and handoffs.
- Measurement of cycle times, queues, and variability.
- Identification of true system constraints.

This isolates the bottleneck or bottlenecks that govern overall throughput.

7. KPI Visibility and Performance Control

We assess whether performance is being actively managed by reviewing:

- Defined KPIs for each major process step.
- Current KPI values versus targets or design intent.
- Visibility, ownership, and decision-making tied to those metrics.

This highlights gaps in measurement that prevent proactive control.

8. Downtime and Stability Analysis

We analyze downtime to determine:

- Pareto drivers of lost capacity.
- Frequency versus duration of failures.
- Recovery time and restart complexity.

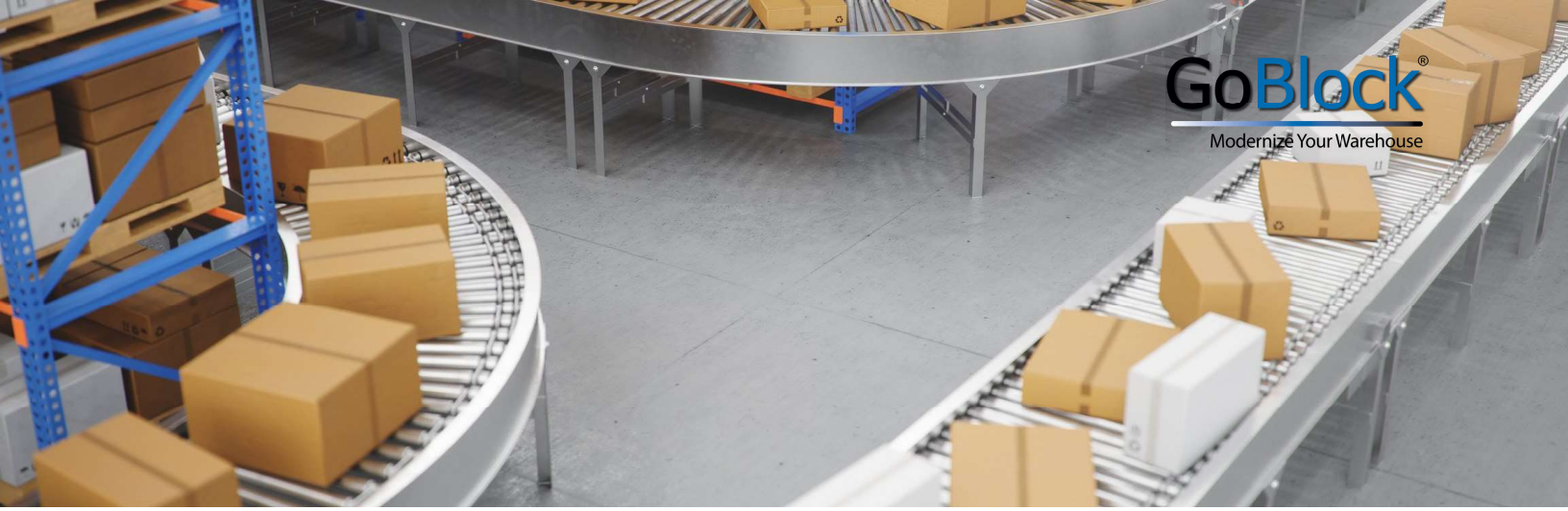
The focus is on identifying the few causes responsible for most performance loss.

9. Labor Model Validation

We compare designed labor assumptions to actual staffing:

- Headcount and skill mix versus plan.
- Labor added to compensate for system limitations.
- Variability across shifts, days, and peaks.

This reveals whether labor shortfalls are structural or symptomatic.



Phase C: Findings and Recommendations

How different is design from reality, and what can we do about it?

The final phase connects intent to reality and translates gaps into prioritized actions.

10. Flow Conformance and Workarounds

We evaluate:

- Where actual material and information flows deviate from design.
- Manual workarounds used to maintain throughput.
- Risks created by reliance on tribal knowledge or informal processes.

This often exposes hidden costs and operational fragility.

11. Financial Impact: Unit Cost, Downtime Cost, and Quality Cost

Operational gaps are translated into financial terms:

- Actual cost per unit produced.
- Cost of downtime, including lost throughput and premium labor.
- Cost of quality, including errors, rework, and service impacts.

This quantification allows leadership to prioritize based on economic impact.

12. Bottleneck Expansion and Improvement Options

With constraints clearly identified, we assess:

- Hardware, software, and operational options to expand capacity.
- Capital versus non-capital improvements.
- Tradeoffs between cost, complexity, and flexibility.

The focus is on realistic, implementable improvements.

13. Support Model Effectiveness

We review whether the current support structure enables rapid issue resolution:

- Vendor, IT, and internal roles and responsibilities.
- On-site versus remote support effectiveness.
- Training, escalation, and accountability.

This determines whether performance issues can be sustained or corrected.



Engagement Deliverables and Outcomes

The engagement concludes with:

- A clear comparison of design intent versus current performance.
- Identification of the primary drivers of underperformance.
- Quantified financial impact of operational gaps.
- A prioritized remediation roadmap distinguishing operational fixes from capital changes.
- Clear guidance on whether additional investment is justified and why.

The result is an objective answer to whether the investment is delivering what was paid for, and a practical plan to improve outcomes where it is not.

Whenever appropriate, the technology vendor can be invited to participate in portions of the engagement to provide technical context, share system expertise, and support the development of practical improvement options.

Engagement Timeline and Responsibilities

Timeline is expected to range from 3-5 weeks, depending on the complexity of the warehouse technology under evaluation.

WEEKS	KEY ACTIVITIES	GOBLOCK LOCATION	CLIENT RESPONSIBILITIES
WEEK 1	**Pre-Engagement:** Kickoff, scope confirmation, stakeholder alignment, and data required. Establish objectives, success criteria, and communication cadence.	Remote	Assign executive sponsor and core contacts. Confirm scope and timeline. Provide access to requested documentation and systems.
WEEKS 1–2	**Phase A: Understanding the Base Design**. Review business case, ROI model, design assumptions, performance specs, contracts, and SLAs.	Remote	Provide business case materials, design documents, contracts, and SLAs. Make stakeholders available for clarification.
WEEKS 2–4	**Phase B: Current State Study** Facility walkthroughs, engineer-led process study, flow observation, downtime review, labor validation, and interviews. Validate bottlenecks, KPI gaps, labor variances, and stability drivers. Compare the current state to the design intent.	Onsite	Provide site access. Ensure availability of ops, IT, and maintenance SMEs. Share historical downtime, labor, and KPI data. Respond to follow-up questions. Validate preliminary observations. Provide missing data as needed.
WEEKS 3–5	**Phase C: Findings and Recommendations**. Quantify financial impacts. Develop remediation options, bottleneck expansion paths, and support model improvements. Draft report review, refinement, and executive report-out. Present conclusions, impacts, and recommended next steps.	Remote or Onsite	Participate in working sessions. Confirm cost assumptions and operational constraints. Ensure executive attendance. Facilitate decision-making, ownership, and next actions.

For discussions, support, or to explore the “Did I Get What I Paid For?” engagement contact GoBlock

contact@goblock.co
or visit www.goblock.co/digwipf

