

Sienda Media

When the Problem Isn't Where It Appears

A Practical Guide to Cross-Functional Business Engineering



Bob Mazzei

Sienda Media | Sienda Ltd

2026

When the Problem Isn't Where It Appears

A Practical Guide to Cross-Functional Business Engineering

Copyright © 2026 Sienda Ltd. All rights reserved.

Published by Sienda Media / Sienda Ltd.

No part of this publication may be reproduced or distributed without permission, except for brief quotations used in reviews or commentary.

This publication provides general business information. It does not constitute legal, tax, financial, certification or other regulated professional advice. Decisions under regulatory and certification frameworks belong to the relevant competent authorities, regulators and certification bodies.

Examples inspired by professional experience have been anonymised, combined or adapted. They do not disclose confidential client information and should not be read as exact accounts of individual assignments.

Contents

The Problem You Were Given	4
Symptoms Are Not Causes	6
Follow the Consequences Across the Business	9
When Commercial Success Creates Operational Failure	12
When Software Gets Blamed	15
When Compliance Reveals the Real Business	18
Project Failure Often Starts Before the Project	21
Find the Decision That Started the Chain	24
Ownership Handoffs and Accountability	27
An Anonymised Business Case	30
A Practical Cross-Functional Diagnostic	35
Solve the Real Problem	41
Where to Look Next	43

The Problem You Were Given

A managing director says that sales are weak. An operations manager says that the new system does not work. A quality manager says that certification is at risk. A project sponsor says that delivery has slipped. Each statement may be accurate, but none should be accepted as a diagnosis. It is a report from one position inside the business. It identifies where pain has become visible, not necessarily where it began.

This distinction matters because organisations tend to solve the problem that can be named most easily. Sales are falling, so salespeople receive new targets. Orders are late, so the warehouse is told to work faster. Data are unreliable, so a new application is purchased. An audit finds missing evidence, so someone is asked to complete the forms. These responses can produce activity without changing the conditions that created the failure.

The problem presented by management should normally be treated as a hypothesis. It is the first reasonable explanation, based on the evidence available to the person raising it. Investigation begins by testing that explanation against the way work actually moves through the organisation. What happened before the symptom appeared? Which decisions changed demand, capacity, timing or responsibility? Where did information pass from one team to another? What did the formal procedure require, and what did people do when pressure increased?

Business engineering is useful in this setting because it does not begin with a preferred department, technology or management fashion. It examines the business as a connected operating system. Commercial promises affect production plans. Production decisions affect quality and fulfilment. Data entry affects purchasing, invoicing and traceability. Governance affects whether exceptions remain temporary or quietly become the normal way of working.

This way of thinking is neither mysterious nor proprietary. It combines observation, process analysis, commercial judgement and disciplined questioning. Its purpose is to establish the chain between a visible symptom and the decisions, assumptions, constraints and handoffs that produced it.

The method is practical. Define the symptom precisely. Reconstruct what changed. Follow the work across departmental boundaries. Compare the recorded process with the real process. Identify the constraint that the organisation has ignored. Then locate the decision that started or sustained the chain.

The word cause should also be used carefully. A business failure rarely has one isolated cause. More often, several conditions combine. An ambitious sales commitment may be manageable when staffing is stable, equipment is reliable and suppliers perform as expected. The same commitment can become damaging when expansion work removes capacity, temporary staff lack training and data arrive late. The task is not to find a convenient person to blame. It is to identify the combination of decisions and conditions that made the outcome likely.

The chapters that follow show how to conduct that investigation. They cover commercial commitments, operations, technology, compliance, projects, ownership and information flows. They also develop an anonymised case in which poor seasonal sales turn out to be the final commercial expression of a much wider operating failure.

The first discipline is simple: resist the urge to prescribe before you have diagnosed.

Symptoms Are Not Causes

A symptom is the point at which a problem becomes observable. It may be a late order, a rejected delivery, a missed margin, a customer complaint, a failed control or an unreliable report. A cause is a condition or decision that helps explain why the symptom occurred. The two can sit in different departments and at different points in time.

Suppose a distributor loses a major customer after repeated late deliveries. The sales director sees lost revenue and describes a retention problem. The account manager may appear responsible because the customer relationship sits with sales. Investigation could show that the warehouse dispatched on time, but production released finished goods late. Production might then point to material shortages. Procurement might show that the materials were ordered against the approved forecast. The forecast, however, excluded a promotion that sales had promised without updating demand planning. The lost account is the symptom. The unvalidated commitment and broken planning handoff are nearer the cause.

Symptoms attract attention because they carry consequences. Causes often look ordinary while they are developing. A missing field in a quotation template does not look serious. Nor does an informal spreadsheet kept by one experienced employee. But if the missing field prevents operations from seeing a customer's packing requirement, and the spreadsheet contains the only accurate conversion between sales units and production units, those small weaknesses can determine whether an order is profitable and deliverable.

Several distinctions improve diagnosis.

First, separate the event from the pattern. One late shipment may result from a vehicle breakdown. A recurring cluster of late shipments on high-volume orders suggests a capacity, scheduling or approval problem. The response should fit the pattern supported by evidence.

Second, separate trigger from cause. A new customer may trigger overload, but the underlying cause may be that capacity had never been measured. A software upgrade may expose duplicate product codes, but the upgrade did not create years of inconsistent master data.

Third, separate errors from the environment. An employee may enter the wrong quantity. If the interface uses an unfamiliar unit, training is incomplete and no validation catches the error, the individual mistake belongs to a wider system of work.

Fourth, separate immediate correction from durable correction. Re-entering a missing batch number may release one shipment. It does not explain why the number was absent, why the omission was not detected earlier or whether other records have the same weakness.

Short scenario

A service company notices that project margins have deteriorated. Finance reports excessive labour cost and asks project managers to control hours. A review of completed work shows that teams are spending unpaid time clarifying requirements after contracts are signed. Sales proposals use broad descriptions because customers want quick prices. Delivery teams then discover exclusions, dependencies and approval needs during implementation. Timesheets reveal the symptom; the commercial definition of work is the stronger point of intervention.

Warning signs

- The proposed solution matches the department reporting the problem without evidence that the cause sits there.
- The explanation depends on labels such as poor communication, resistance to change or lack of accountability without identifying a specific failed decision or handoff.
- Management asks who made the mistake before establishing how the process allowed it.
- The same symptom returns after repeated local fixes.

→ Reports improve, but customers, cash, quality or delivery do not.

Diagnostic questions

- What exactly failed, for whom, and when?
- Is this a single event or a recurring pattern?
- What changed immediately before the first occurrence?
- What had to happen upstream for this result to become possible?
- Which control should have detected the problem earlier?
- Did the control exist only on paper, or was it used in practice?

An accurate symptom statement is narrow and observable: twenty-seven per cent of priority orders missed the agreed dispatch date during a defined period. A weak statement contains a conclusion disguised as a fact: the warehouse lacks commitment. Begin with what can be verified. Explanations come later.

Follow the Consequences Across the Business

Departments are useful for organising people, budgets and specialist knowledge. They are poor maps of how value reaches a customer. An order can begin with marketing, take shape in sales, become a requirement for planning, draw materials through procurement, consume production and quality capacity, generate records in several systems and finally affect cash. A decision made in one place can alter work everywhere else.

Cross-functional analysis follows consequences rather than organisation charts. Start with an important transaction or commitment and trace what each function must do because of it. A new product, for example, may require a new supplier, revised specifications, different storage, staff training, amended labels, new product codes, quality checks and customer documentation. Commercial approval alone does not make the product operationally ready.

A useful device is the consequence chain. Write the original decision at the left. Add each operational consequence in sequence. For every link, record the owner, information required, capacity consumed, control applied and evidence produced. The chain exposes missing work that a departmental plan may hide.

Consider a manufacturer that reduces minimum order quantities to win smaller customers. Sales sees a broader market and more orders. Planning receives more frequent demand with greater variation. Production performs more changeovers. Quality conducts more start-up checks. Warehousing handles more picking lines. Finance processes more invoices with a lower average value. Customer service manages more delivery enquiries. The policy may still be sound, but its profitability cannot be judged from the sales increase alone.

Consequences also travel backwards. A slow approval in quality may force sales to delay a launch. A procurement rule may determine which promises operations can make. A finance decision to delay supplier payment may reduce the willingness of critical suppliers to accept urgent orders. The flow is a network, not a neat sequence.

The investigator should look for four types of propagation.

Volume propagation occurs when more transactions create more work elsewhere. Ten per cent more orders can create far more than ten per cent more handling if order sizes fall or product variation rises.

Timing propagation occurs when one team consumes time needed by another. A quotation approved late may compress procurement and production without moving the customer's delivery date.

Variation propagation occurs when exceptions multiply. Special labels, unusual payment terms and customer-specific packing can each be reasonable. Together they can destroy standard work.

Information propagation occurs when a fact must be preserved across systems and teams. If a customer requirement begins in an email and must later appear in an ERP order, production instruction and delivery record, every transfer creates risk.

Short scenario

A professional-services firm offers fixed-price work to simplify buying. Revenue grows, but cash becomes erratic. The work requires client documents before delivery can begin, yet contracts do not define submission deadlines. Projects remain open, staff capacity stays reserved and invoices wait for milestones that the firm cannot reach. Finance experiences a cash problem, delivery experiences a scheduling problem and sales experiences pressure to close more work. The common cause is the commercial design of the offer.

Follow the work checklist

- Choose one real order, project, batch or customer request.
- Identify every team that touches it, including finance and quality.
- Record what each team receives and what it must produce.
- Note every change in format, unit, system or owner.

- Mark queues, approvals, rework and informal workarounds.
- Compare promised time with actual elapsed and working time.
- Identify where an exception became routine.

The aim is not to produce an impressive process map. The aim is to understand causality well enough to make a decision. Stop mapping when the evidence explains the symptom and identifies a controllable intervention. Endless detail can become another way to avoid judgement.

When Commercial Success Creates Operational Failure

Winning business is not the same as being able to deliver it. A sale creates an operational obligation. Volume, delivery date, service level, specification, packaging, reporting and payment terms all translate into work and risk. If those terms are accepted without operational validation, apparent commercial success can damage service, margin and reputation.

The danger is greatest when incentives and information are divided. Sales is rewarded for revenue. Operations are measured on efficiency. Procurement is measured on cost. Quality protects standards. Finance protects margin and cash. Each function may act rationally within its own target while the complete outcome deteriorates.

Five commercial commitments deserve explicit validation.

Volume requires evidence of usable capacity, not theoretical capacity. A line rated for a certain output under ideal conditions does not have that capacity when changeovers, cleaning, maintenance, staff availability and quality holds are included.

Deadline requires a realistic lead-time calculation. The customer's requested date cannot simply be passed downstream as an instruction. Time already consumed in negotiation and approval must be visible.

Service level requires a clear definition and cost. A same-day response, emergency replenishment or guaranteed availability can create standby capacity that is invisible in the selling price.

Specification requires control of variation. A small change in label, tolerance, reporting or packing may force a separate run or manual check.

Price requires a cost model that reflects the actual fulfilment route. Standard cost is misleading when the order needs overtime, expedited materials, small batches, rework or external storage.

An operational validation does not need to become a committee. It needs defined thresholds. Routine commitments can pass automatically. Exceptions involving unusual volume, dates, specifications or terms should trigger review by the functions that will carry the consequence.

Short scenario

A food producer secures a seasonal listing with a national customer. The headline volume appears attractive. The customer requires narrow delivery windows, mixed pallets and promotional labels that change during the campaign. The business prices the deal using normal packing speed and standard pallet handling. During the season, label changes cause stoppages, mixed pallets occupy staging space and missed collection windows lead to urgent transport. Revenue rises while contribution falls. The problem is described as factory inefficiency, but the factory is executing a commercial design that was never costed as sold.

Warning signs

- Sales forecasts are discussed, but capacity assumptions are not recorded.
- Customer-specific requirements remain in emails rather than controlled order data.
- Delivery dates are confirmed before procurement or production reviews constraints.
- High revenue customers generate repeated expedites and credits.
- Overtime is treated as evidence of commitment rather than evidence of unstable planning.
- No one can show the contribution of an order after exceptions and fulfilment costs.

Questions before accepting an exception

- Which standard operating condition changes?
- What extra work, delay or control follows?

- Which constrained resource will the order consume?
- What other work may be displaced?
- Has quality confirmed that the specification is achievable and verifiable?
- Has finance tested the margin and cash effect using the actual route?
- Who has authority to accept the residual risk?

The objective is not to allow operations to veto growth. It is to make the promise consciously. A good commercial process distinguishes ambition from assumption. Ambition sets a demanding outcome. Assumption pretends that the capabilities required to achieve it already exist.

When a business repeatedly sells beyond its operating design, employees create workarounds. Planners change sequences. Buyers expedite. Quality staff inspect late. Warehouse teams relabel. Account managers negotiate apologies. These actions can keep customers supplied for a time, but they also conceal the true cost of the commitment. Management sees heroic recovery rather than a structurally unprofitable promise.

The remedy begins upstream. Define which terms require cross-functional approval. Make lead times and constraints visible during quotation. Record customer requirements once, in a controlled form. Review realised margin after delivery, not only quoted margin before it. Commercial energy remains valuable; it simply operates with a complete view of the business it commits.

When Software Gets Blamed

Software is a convenient defendant. It has a name, a supplier and a screen that everyone can criticise. Processes are harder to confront because they belong to the organisation itself. When a system appears to fail, the first task is to separate defects in the technology from defects in the operating design around it.

A genuine software defect means the system behaves contrary to its specified and intended operation. Many complaints do not meet that test. The system may lack necessary data, receive inconsistent data, reflect contradictory rules or be used through an improvised sequence that nobody designed. It can still be unsuitable, but the diagnosis is different.

Technology amplifies the process it is given. A stable process with clear ownership becomes faster and more visible. An unstable process becomes a faster producer of inconsistent records. Automation can conceal this for a while because transactions move without obvious human delay. The contradiction appears later in stock, invoices, reports or customer service.

Begin with one disputed transaction. Ask the user to demonstrate what happened, from the first business event to the final output. Observe the work rather than relying only on descriptions. Note which fields come from customers, which are retyped, which default automatically and which are changed outside the system. Look for spreadsheets, messaging threads, paper notes and personal memory. These are not trivial side issues. They often contain the real process.

The distinction between system of record and source of truth is important. An ERP may be called the system of record, yet employees may trust a planner's spreadsheet because it contains late changes. The ERP then records an authorised version of events while the spreadsheet directs actual work. Reports can be internally consistent and operationally false.

Short scenario

A wholesaler reports that its stock system is unreliable. Physical counts differ from recorded quantities. The software provider finds no calculation error. Observation shows that urgent customer collections bypass the normal picking queue. Staff write quantities on paper and enter them later. When several customers arrive together, papers are combined and product units are confused with cases. The system is blamed because the discrepancy appears there. The failure begins when the urgent route separates physical movement from immediate recording.

Another common pattern involves responsibilities. A field is mandatory in policy but optional in software because no department accepted ownership of it. Sales expects operations to complete it. Operations believes the information should come from sales. Users insert placeholders to continue. Months later, management commissions a dashboard and discovers that the field cannot support analysis. The dashboard is not the source of poor data. It exposes an unresolved handoff.

Diagnostic questions for an apparent software failure

- What business event should the system represent?
- Who first knows the required information?
- When and where is it recorded?
- Is the same fact retyped or transformed later?
- What do users do when the normal route is too slow or incomplete?
- Which rules differ between teams, customers or sites?
- What validation prevents an impossible or incomplete entry?
- Does the report reflect the process management believes exists or the one employees actually use?

- Is the requested change correcting a defect, encoding a policy decision or compensating for weak discipline?

Warning signs

- Requirements are expressed as screens and buttons without describing the business decision they support.
- Users ask for automation before agreeing on definitions.
- The same customer, product or supplier has several identifiers.
- Critical work depends on one person's spreadsheet.
- Staff complete records after the physical event.
- Management asks for a new dashboard while source data remain disputed.

Technology can still be the correct intervention. The process may require validation, integration, simpler interfaces or better exception handling. But the specification should follow diagnosis. Otherwise the organisation pays to automate ambiguity. A new system then inherits the old contradictions, and the project is judged against expectations that no software could satisfy.

The most useful test is simple: if the software disappeared tomorrow, how would the work have to be performed correctly? That question reveals the business rules, roles and evidence the system must support. It also reveals whether those rules have ever been agreed.

When Compliance Reveals the Real Business

An audit does more than test documents. It compares the organisation's stated controls with evidence of what occurred. The resulting gaps often reveal weaknesses that affect customers, cost and continuity long before they create a formal compliance consequence.

Certification and regulatory frameworks differ, and decisions belong to the relevant competent authorities, regulators or certification bodies. This chapter does not provide legal advice or promise certification outcomes. It examines the operational value of readiness work: clear processes, defined responsibility, reliable records, traceability and controlled exceptions.

A missing record is rarely only a paperwork problem. It can mean that nobody performed the control, that the control occurred but was not recorded, that evidence sits in an inaccessible place or that the process design makes compliance unrealistic during busy periods. Each explanation requires a different response.

Suppose a traceability exercise cannot connect finished goods quickly to incoming material. Management may ask quality staff to improve record keeping. Investigation could reveal that production combines partial quantities from several lots to avoid waste, while the ERP expects one input lot per batch. Operators therefore add details in free text. The weakness involves process design, system configuration, production economics and quality control. Asking quality to write a new procedure will not reconcile them.

Compliance work is especially useful because it tests governance under pressure. A procedure describes the normal route. The real test is what happens when a supplier is late, a customer changes an order, equipment fails or a deadline approaches. Who can approve an exception? What evidence must remain? When does temporary deviation end? If no answer exists, operational urgency becomes the authority.

Frameworks such as GLOBALG.A.P., ISO management systems and privacy controls can expose similar organisational questions even though their scopes differ. Who owns the complete process? Are

responsibilities understood? Can the business show that a control operated? Are external providers governed? Does information remain accurate as it moves? Can the organisation identify affected products, records or people when something goes wrong?

Short scenario

A growing business uses an external provider for part of its processing. Commercial teams treat the provider as extra capacity. Operations treat it as a transport destination. Quality discovers during readiness work that specifications, lot identity, release responsibilities and evidence return have not been defined consistently. The certification concern is real, but it exposes a wider governance failure: the business expanded its operating model without designing the handoff.

Four questions for every control

- Purpose: What risk or requirement does the control address?
- Owner: Who must perform it and who reviews exceptions?
- Evidence: What record proves that it occurred?
- Reality: Can it be performed reliably under normal workload and foreseeable pressure?

Warning signs

- Documents are created immediately before audits and are rarely used in daily work.
- Several versions of a procedure circulate.
- Employees can describe the workaround more clearly than the formal process.
- External providers sit outside internal process maps.
- Corrective actions close when a document is issued, without testing whether behaviour changed.
- Records are complete only when one experienced employee is present.

Compliance should not be used as a threat or as a substitute for management. Nor should the business optimise only for passing an audit. A polished file of procedures can coexist with poor control. Conversely, sound operations can suffer if evidence is not preserved. Operational readiness requires both performance and proof.

When an audit finding appears, trace it in two directions. Look upstream for the decision, resource or information that made the failure likely. Look downstream for the business consequence beyond the audit: rejected product, delayed release, weak investigation, privacy exposure, customer concern or dependence on personal knowledge. This turns compliance from a departmental burden into evidence about how the business is governed.

Project Failure Often Starts Before the Project

Projects often inherit failure before a project manager is appointed. A commercial deadline has already been promised. A budget has already been presented as sufficient. A technology has already been selected. A benefit has already entered the business case. The project is then asked to organise assumptions that were never tested.

This does not mean planning must eliminate uncertainty. Projects exist partly because the organisation is changing something it does not yet fully know. The requirement is to identify material uncertainty and assign decisions before commitments become irreversible.

Five pre-project conditions deserve attention.

The first is an untested problem definition. If management says that the business needs a new system, the project may begin by selecting software instead of establishing what is wrong with the current operation. Success is then measured by installation rather than improved performance.

The second is a deadline derived from desire rather than work. A customer event, financial period or contract date may be genuinely fixed. The team must then reduce scope, add capacity or accept defined risk. Calling every date fixed while preserving full scope and budget merely transfers the contradiction to delivery.

The third is missing operational stakeholders. Sponsors and specialists may design a change without the people who receive orders, schedule work, maintain data, resolve exceptions or support customers. Their knowledge arrives late as resistance or scope creep, although it was always part of the real requirement.

The fourth is a benefit without an owner. A project can deliver a capability, but operational management must change behaviour to realise a benefit. Installing planning software does not improve forecasts if commercial teams still withhold information or product data remain inconsistent.

The fifth is a hidden dependency. A launch may depend on supplier approval, data cleansing, training, site work or a customer decision. If the dependency sits outside the project plan, progress reporting creates false confidence.

Short scenario

A company commits to open an expanded facility before its seasonal peak. The construction schedule, equipment installation, staff recruitment, certification extension and customer trials are managed as separate streams. Each stream reports acceptable progress. Nobody owns the integrated operating date. Equipment arrives, but utilities are not ready. Recruitment follows the original date, so trained staff wait. Customer trials are compressed into the final weeks. The project appears to fail during commissioning; the failure began when the business treated one operating transition as several independent projects.

Questions before authorising a project

- What observable business problem or opportunity justifies the work?
- Which assumptions are critical to cost, time and benefit?
- Who can validate each assumption?
- Which date is genuinely fixed, and what may change to protect it?
- Which operating teams must adopt the result?
- What data, approvals and external decisions are dependencies?
- Who owns benefits after project closure?
- What evidence would justify stopping or redesigning the project?

Scope creep also requires careful diagnosis. Some additions are avoidable preferences. Others are requirements discovered late because initial analysis was narrow. Labelling every discovery as poor change

control can protect the plan while damaging the outcome. The correct question is whether the new requirement changes the agreed purpose or reveals work always necessary to achieve it.

Project governance should surface decisions, not manufacture paperwork. A useful status report states what has changed, which assumption is threatened, what consequence follows and who must decide by when. A weak report lists activities and percentage complete while the central feasibility question remains unresolved.

When a project is struggling, reconstruct its history before replacing people or tools. Review the original commercial commitment, problem statement, assumptions, exclusions and approval sequence. The root cause may pre-date the schedule everyone is now trying to recover.

Find the Decision That Started the Chain

Root-cause investigation is sometimes presented as a hunt for a single hidden answer. Businesses are less tidy. Outcomes emerge from decisions interacting with constraints and later responses. The practical goal is to find the earliest controllable decision that materially changed the chain, together with the conditions that allowed its effects to continue.

Start with a precise symptom and move backwards through evidence. Avoid asking why in the abstract. Ask what happened immediately before the event, what information was available, what decision was made, who had authority and what constraint applied. Continue until the chain reaches a decision that management can change or a condition it must consciously accept.

A simple investigation record has six columns: event, time, decision, information available, owner and consequence. It can be built from orders, system records, emails, schedules, meeting decisions, interviews and direct observation. The purpose is not courtroom certainty. It is a reliable account strong enough to guide action.

The evidence sequence usually passes through four layers.

Define the symptom. Specify the affected customer, product, project, process or period. State the difference between expected and observed performance.

Reconstruct the timeline. Identify what changed before the problem appeared. Include demand, staffing, suppliers, equipment, rules, systems, projects and commercial terms.

Trace decisions and assumptions. Determine what people decided, what they believed and which information they lacked or ignored.

Test alternative explanations. Look for cases where the symptom did not occur. Differences between successful and failed transactions often reveal the condition that matters.

Short scenario

A business receives more invoice disputes after introducing a new pricing tool. It is tempting to blame calculation logic. A transaction review shows that disputed invoices mainly involve contracts renewed before the tool launched. Renewal letters contain negotiated exceptions that were never converted into structured contract data. The tool calculates correctly from incomplete inputs. The relevant decision occurred months earlier, when the renewal campaign used free-text documents without a controlled route for updating billing rules.

Beware of stopping at human error. Saying that a buyer forgot to order, an operator selected the wrong code or a project manager missed a dependency describes the last visible action. Ask what made the error possible and consequential. Was the workload unreasonable? Were codes ambiguous? Did the system permit an impossible choice? Did the review occur too late? Had responsibility been split without a final owner?

Beware also of infinite depth. Investigation can always move further into culture, incentives or history. Stop when four conditions are met:

- The explanation accounts for the observed pattern.
- Evidence distinguishes it from plausible alternatives.
- The organisation can act on it.
- The proposed action would reduce recurrence rather than only repair the current case.

The counterfactual test is useful. Ask: if this decision had been different, while other conditions remained, would the failure probably have occurred? If changing the decision would not alter the outcome, it is not a sufficient intervention point.

The recurrence test then asks: if we apply the proposed fix, what prevents the same chain in the next busy period, customer exception or staff absence? Training one person may correct knowledge. It will not correct an approval design that routinely bypasses that person.

Root-cause work should produce an action hierarchy. Contain the immediate consequence. Correct the affected transaction or record. Change the process or decision rule. Assign ownership and evidence. Monitor whether the symptom and its precursors decline. Each action answers a different need; confusing them leads to premature closure.

The final explanation should fit on one page. It should state the symptom, causal chain, supporting evidence, rejected alternatives, immediate containment, durable correction and owner. If the explanation requires dense jargon, the thinking is probably unfinished.

Ownership Handoffs and Accountability

Many serious failures occur between departments. Each team completes its assigned task, yet the overall outcome fails. Sales enters the order, planning schedules it, production makes it, quality checks it and logistics dispatches it. If the customer's special requirement changes meaning between those stages, local completion offers little protection.

A handoff is any point where responsibility, information, material or authority moves. Handoffs create risk because context can be lost. The sender assumes the receiver understands. The receiver assumes the sender validated. Systems use different fields or units. Deadlines leave no time to clarify. Exceptions travel through informal channels because the formal route cannot represent them.

Good handoff design answers five questions. What must be transferred? In what form? By when? What makes it acceptable? Who acts when it is incomplete? Without these rules, teams compensate through relationships and memory. This can work while experienced people remain available and volume is low. Growth exposes weakness.

Process ownership is different from departmental responsibility. A department owns its work. A process owner is accountable for the outcome across the full route. The owner does not perform every task or command every participant. The owner ensures that the stages fit together, measures the complete result and has authority to convene decisions when local targets conflict.

Consider order fulfilment. Sales may own quotations and order entry. Operations may own production. Logistics may own dispatch. Finance may own credit and invoicing. The complete outcome is a correctly specified, profitable order delivered as promised and invoiced accurately. If nobody owns that outcome, every delay becomes another department's problem.

Short scenario

A customer requests a change to packaging artwork. The account manager obtains approval and forwards an email to planning. Planning updates the schedule but assumes procurement received the change. Procurement orders the old material because the approved specification repository has not changed. Quality stops the run. Every individual can show a reasonable action, yet the handoff design has no controlled mechanism linking customer approval, specification change, material order and production release.

Accountability is weakened by vague collective language. Statements such as the team will monitor, the business will ensure or stakeholders will review and do not identify who must act. For each control, name one accountable role, even when several roles contribute. Accountability should follow authority; assigning an owner who cannot change the process is merely to blame in advance.

Handoff review

- Identify the exact item transferred: data, material, approval, instruction or responsibility.
- Name the sending and receiving roles.
- Define the acceptance criteria.
- Record the system or controlled document used.
- Specify the time available for review.
- Define the exception route and decision authority.
- Measure failure at the handoff, not only at the final outcome.

Warning signs

- Important instructions are forwarded rather than entered into a controlled workflow.
- Two departments each believe the other validates the same field.
- Meetings resolve recurring exceptions without changing the process.
- Performance measures reward local speed while the total lead time worsens.
- Escalation depends on personal access to a senior manager.
- The process owner is named but lacks data or decision rights.

Controls should sit as near as possible to the point of risk. If sales first knows a required delivery temperature, validation should occur during order capture, not when logistics loads the vehicle. If production changes a lot, traceability should update at the physical event, not at the end of the shift. Late controls are expensive because they detect failure after more work has accumulated.

Accountability does not mean creating approval gates everywhere. Excessive approval drives work underground. The design should give routine work a clear path and isolate genuine exceptions. The owner then focuses on patterns: why exceptions occur, what they cost and whether the standard process must change.

An Anonymised Business Case

The following case has been anonymised and adapted for confidentiality. It combines and modifies operational details and should not be read as an exact account of any client assignment. Companies, individuals, locations, dates, quantities and financial information have been removed or changed.

The presented problem

A fresh-produce business ended an important season below its sales plan. Management described the issue as weak commercial performance. The sales team had forecast strong demand from established retail and wholesale customers, yet realised sales fell short. Several customers reduced orders during the peak weeks, and part of the crop was sold later through lower-value channels.

The initial proposals were familiar: replace an account manager, intensify customer visits, revise pricing and recruit an additional salesperson before the next season. Those actions assumed that customers had not bought enough. The first review therefore separated orders not won from orders reduced, delayed, rejected or diverted after commitment.

The pattern looked different. Early interest had been healthy. The largest gap arose after the business began supplying the season. Customers reduced subsequent orders following inconsistent deliveries and several quality disputes. The commercial team was reporting the final consequence of an operating failure.

Commercial commitments

Before the season, the business had pursued larger programmes to support an expansion plan. Commercial discussions assumed that new growing capacity and improvements to the packing operation would be available by the peak. Customers received confident volume indications, narrow delivery schedules and several packing options.

The commitments were not reckless in isolation. Each depended on a reasonable assumption: crop availability, completion of expansion work, normal packing performance and access to temporary labour. The weakness was that nobody had tested the assumptions together. The sales forecast became the production expectation, but it did not show which capacity and timing conditions had to hold.

Production and project pressure

Expansion work competed with seasonal preparation. Managers who normally checked labour plans, maintenance and supplier readiness were occupied with contractors and commissioning. Some improvements finished later than expected. Existing areas remained usable, but movement through the site became less direct and storage space was constrained.

Production volume was sufficient in broad terms. The problem was the usable volume by day, grade and packing format. Harvest plans followed crop condition and field efficiency. Customer programmes required specific products at specific times. The difference created peaks that the packing operation could not absorb without holding products or changing sequence.

The expansion project reported physical completion tasks. Seasonal readiness was managed separately. No one owned the combined question: can the expanded operation receive, cool, pack, identify and dispatch the promised mix at the required rate on the required dates?

Packhouse operations

As pressure increased, supervisors used practical workarounds. Product was staged wherever space became available. Packing sequences changed to protect urgent orders. Part-completed lots waited while lines switched format. Temporary staff were assigned through verbal instructions because controlled job information was slow to update.

These choices kept products moving, but they increased travel, handling and waiting. The nominal line speed remained unchanged, so performance reports did not explain the lost time. The constraint was not simply the packing machine. It was the complete flow through intake, cooling, staging, packing, checking and dispatch.

Cold-chain shortcuts

Cooling capacity and staging discipline became critical during peak intake. Products sometimes waited longer than planned before cooling. At other times, cooled products moved into warmer staging areas while teams assembled urgent orders. Temperature records existed, but they described locations and scheduled checks more reliably than the history of each lot.

No single shortcut appeared decisive. The cumulative effect was greater variation in product condition on arrival. Customers experienced the variation as inconsistent shelf life. Complaints were initially debated as matters of product quality or customer handling. The investigation connected them with timing and movement through the packhouse.

Traceability and ERP data

The ERP expected a relatively orderly relationship between incoming lots, packing runs and finished product. Operational workarounds created splits, combinations and sequence changes. Staff recorded some changes after the event. Free-text notes and paper records supplemented the system. The final records could often be reconciled, but not quickly and not always without the knowledge of the supervisor who had managed the shift.

This produced two consequences. First, managers could not compare operational conditions with complaints reliably. Second, investigations took longer, which weakened the business's response to customers. The ERP was described as cumbersome, yet its difficulty reflected a process that had moved beyond the assumptions on which data capture was based.

Certification controls

Readiness reviews then exposed related weaknesses. Procedures described standard product flow, lot identification and temperature control. They did not fully describe the temporary staging and mixed-lot practices used under pressure. Corrective activity focused at first on completing records and reminding staff of procedures.

That response was insufficient. The operating process and the documented control system no longer matched. A competent certification body would make its own decisions under the applicable scheme; the business task was to restore operational control and reliable evidence, not to predict or influence that decision.

Commercial rejection

Several deliveries were disputed or rejected. Others were accepted with credits. More damagingly, customers reduced later programmes because supply felt less dependable. Sales staff spent time managing complaints and renegotiating volumes rather than developing accounts. Lower realised sales were therefore not evidence of weak market effort. They were the commercial consequence of variability created upstream.

The causal chain

The expansion objective encouraged larger commercial commitments. Those commitments assumed combined capacity that had not been validated. Project pressure reduced seasonal preparation and constrained the operating layout. Peaks created packhouse workarounds. Those workarounds weakened cooling discipline and the immediate capture of lot movements. Variable product conditions and slow investigations reduced customer confidence. Customers then rejected, credited or reduced orders. The symptom appeared in the sales report.

No link alone explains the entire outcome. Weather, crop variation and customer decisions still mattered. The evidence nevertheless showed a controllable chain that management could address.

The corrective plan

Immediate containment focused on product flow, temperature discipline, lot identification and customer response. The business limited certain packing combinations during peak periods and made exception approval explicit.

Before the next season, commercial programmes were translated into daily capacity and format requirements. Operations validated commitments above defined thresholds. The expansion plan gained an integrated readiness review covering people, layout, equipment, data and controls. ERP changes supported real lot movements rather than the idealised route. Traceability exercises used difficult peak scenarios, not only standard examples.

Management also changed the measure of seasonal performance. Revenue remained important, but it was reviewed with order reliability, credits, rejected quantities, realised margin and time spent on exceptions. This prevented sales growth from concealing operational cost.

Lessons from the case

1. The department that reports the loss may be furthest downstream from its cause.
2. Capacity must be tested against timing, product mix and control requirements.
3. Project completion and operational readiness are different decisions.
4. Workarounds are evidence about process design, even when they are necessary.
5. Data weakness often begins where physical work and formal recording separate.
6. Compliance findings can reveal commercial risk before management sees the financial pattern.
7. Customer confidence depends on dependable operation and credible investigation, not only relationship management.

The central management error was not one bad decision. It was approving several connected decisions separately. Cross-functional diagnosis restored the connections.

A Practical Cross-Functional Diagnostic

The following diagnostic is designed for an owner, director or senior manager facing a material business problem. It can be used in a focused review lasting several days or as the structure for a wider investigation. The scale changes; the logic does not.

Step 1 Define the symptom

Write one observable statement. Identify what happened, where, when and against which expectation. Avoid blame and explanation.

Weak statement: Operations cannot cope.

Useful statement: During the last six weeks, twelve priority orders missed the confirmed dispatch date, compared with two during the preceding six weeks.

Record who experiences the consequence: customer, employee, supplier, regulator, certification body, shareholder or another part of the business. Estimate the operational and commercial exposure without inventing precision.

Step 2 Establish what changed

Look immediately before the first occurrence and across the period in which the pattern developed. Changes may include customers, products, volume, mix, prices, suppliers, staffing, equipment, layouts, systems, policies, incentives or projects.

Do not assume the most visible recent change is causal. Record candidates, then test them against successful and failed cases.

Step 3 Select real transactions

Choose a small set of orders, projects, batches, complaints or records that represent the problem. Include at least one comparable case that succeeded. Follow evidence from beginning to end. Aggregate reports show patterns; transactions show mechanisms.

Step 4 Map the consequence chain

For each case, record the original commitment and every function it affected. Identify volume, timing, variation and information consequences. Note where a team received less time, incomplete data or an unusual requirement.

Step 5 Compare formal and actual work

Read the procedure, system design or policy. Then observe the work and ask employees to demonstrate it. Record differences without immediate judgement. A workaround may be unsafe, but it may also reveal a legitimate requirement that the formal process ignores.

Step 6 Examine handoffs

At every transfer, ask what moves, who sends it, who receives it, how acceptance is shown and what happens when information is incomplete. Pay particular attention to changes of system, unit, format and responsibility.

Step 7 Identify assumptions and constraints

Write the assumptions behind the relevant commercial or management decision. Then list the constraints: usable capacity, lead time, skills, cash, supplier reliability, space, approval, data quality, equipment or control requirements. Determine which constraint was ignored, hidden or changed.

Step 8 Trace decisions backwards

Build the event, decision, information, owner and consequence record. Continue backwards until you reach an intervention point that explains the pattern and sits within management control.

Step 9 Test the explanation

Ask whether the proposed cause explains why some cases failed and others succeeded. Seek disconfirming evidence. Use the counterfactual test: if the decision or condition had differed, would the result probably have changed?

Step 10 Design layered action

Separate containment from correction. Containment protects the current customer, product, project or record. Correction changes the process, decision rule, system, capability or control. Assign one accountable owner and define evidence that the correction operates.

Step 11 Monitor precursors

Final outcomes arrive late. Identify earlier signals in the chain: unapproved exceptions, missing fields, queue age, delayed cooling, repeated schedule changes, unmatched records or work awaiting clarification. Monitoring precursors gives management time to act before revenue, quality or customers are affected.

The core questions

- What changed immediately before the problem appeared?
- Which department sees the symptom?
- Which department controls the underlying process?
- What assumptions were made?
- What commitments were made?

- Where does information change hands?
- Where are people bypassing the formal process?
- What does the system record versus what actually happens?
- Which constraint is being ignored?
- Who owns the complete outcome?
- What happens upstream?
- What happens downstream?

Evidence to request

- A sample of real transactions, including successful cases.
- Original customer, supplier or project commitments.
- Forecasts and capacity assumptions used at the time.
- Process instructions and approval rules.
- System records, change histories and exception notes.
- Measures of waiting, rework, credits, rejection and expedite cost.
- Corrective actions from earlier occurrences.
- Interviews with people who perform and receive the work.

Interview discipline

Ask people to describe the last real occurrence, not the process in general. Questions such as show me what you received, what did you do next and what happens when this field is missing, produce better

evidence than asking whether communication is effective. Interview functions separately before bringing them together. This reduces the tendency to repeat the accepted organisational story.

Diagnostic warning signs

- The team agrees on a solution before agreeing on the symptom.
- Every explanation ends at the boundary of the speaker's department.
- Senior commitments are treated as facts rather than decisions that can be tested.
- The process map contains no exceptions, queues or rework.
- Employees fear that describing actual work will be treated as misconduct.
- The system contains clean data only after manual reconciliation.
- Corrective actions add checks but do not remove the condition producing errors.
- Nobody owns the outcome from customer promise to cash and evidence.

A concise diagnostic report

The final report should be short enough to drive a decision. It should contain:

- The precise symptom and scope.
- The causal chain and material contributing conditions.
- Evidence that supports the explanation.
- Alternative explanations tested and why they were rejected or retained.
- Immediate containment.
- Durable corrective actions, owners and dates.

- Measures that will show whether the chain has changed.
- Open risks requiring specialist, regulatory or certification judgement.

The report should distinguish what is known, what is inferred and what remains uncertain. Senior judgement is valuable partly because it can act without pretending uncertainty has disappeared.

The diagnostic is complete when management can see how the result was produced, where intervention is possible and who will own the complete outcome. A longer report is not a stronger report. The strength lies in the causal account and the quality of the decisions that follow.

Solve the Real Problem

Organisations naturally respond where pain is visible. A customer leaves, a system produces bad data, an audit raises a finding or a project misses a date. Pressure demands action. The risk is that urgency turns the first explanation into an accepted diagnosis.

Cross-functional business engineering introduces a deliberate pause between symptom and prescription. It asks management to define the observed failure, reconstruct what changed, follow consequences across the business and compare formal work with actual work. It traces the chain through commitments, assumptions, constraints, decisions and handoffs until it reaches a point where action can reduce recurrence.

This approach often changes the ownership of the response. A sales problem may require an operational correction. A software complaint may require agreement on data and responsibility. A certification concern may require governance of an external process. A project recovery may require revisiting a promise made before the project existed.

The purpose is not to shift blame to another department. It is to replace departmental explanations with an account of how the business produced the outcome. That account makes responsibility clearer because it identifies decisions and controls, not personalities.

Good diagnosis also protects investment. It reduces the chance that a business buys technology to encode ambiguity, adds staff to compensate for poor flow, discounts price to offset unreliable service or creates paperwork to conceal weak control. The correct intervention may still involve software, recruitment, training or a new procedure. Diagnosis determines what those measures must change.

The practical discipline can be remembered in six actions: define the symptom, reconstruct the change, follow the transaction, examine the handoffs, test the causal chain and assign the complete outcome. Use

evidence from real work. Include successful cases. Treat workarounds as information. Separate containment from correction.

Management will never possess perfect information. It can still insist on an explanation that fits the evidence, acknowledges uncertainty and crosses organisational boundaries. That is the foundation of sound judgement.

Before asking how to fix the problem, make sure you know which problem you are actually fixing.

Where to Look Next

Use this table to broaden the first investigation. It does not replace evidence from real transactions.

Visible symptom	Investigate across	Decisive question
Falling sales	Order reliability, availability, quality variation, rejected deliveries, customer credits	Which promised conditions failed after the order was won?
Poor margin	Actual fulfilment route, changeovers, rework, expedites, small batches, procurement	Which costs arise only after the commercial commitment?
Late delivery	Promise date, planning queue, material availability, approvals, product mix	Who consumed the time before operations received the work?
Bad data	Source event, definitions, re-entry, workarounds, ownership, validation	What happens in the business before the record reaches the system?
Software complaints	Process variation, unclear rules, master data, exception routes	Is the system defective, unsuitable, or faithfully recording a broken process?
Audit finding	Daily practice, ownership, evidence, exception governance, external providers	What operating weakness does the missing evidence reveal?
Project delay	Pre-project promise, dependencies, feasibility, decision latency, adoption	Which assumption was fixed before the project tested it?

About Sienda

Sienda Ltd provides independent business engineering and advisory services involving business processes, commercial decisions, projects, technology and decision analysis. Its work helps organisations understand how decisions and operating conditions combine across functions, so that management can address causes rather than symptoms.

About Bob Mazzei

Bob Mazzei has nearly four decades of cross-functional business experience spanning business analysis, commercial development, operations, project and product management, technology and professional training. He has worked internationally with organisations in the United Kingdom, Italy, Spain, the United States and wider global markets, across sectors including agrifood, healthcare, legal services and information technology.

sienda.co.uk