Rebuilding Support from the Ground Up

Three Assumptions AI Gives Us Permission to Break

Omid Razavi · July 19, 2026

Most support organizations are still applying AI to an operating model built around cases, queues, owners, handoffs, escalations, and customer follow-up.

They summarize tickets faster. They improve routing. They help agents find answers. They reduce some manual work.

That is useful, but it leaves the basic architecture untouched.

The larger opportunity is to redesign support around what is now possible: continuous telemetry, AI-led investigation, product intelligence, decision orchestration, and human judgment applied where it creates the greatest value.

I recently explored this with Avi Karnon of Cribl. Our discussion went well beyond incremental automation and challenged assumptions that have shaped technical support for decades.

Three deserve to be broken.

1. Every case needs one persistent human owner.

2. The customer must remain central to the investigation.

3. Support should begin only after the customer reports a problem

These assumptions made sense when knowledge lived mostly in people, diagnostic access was limited, and support could only react after a customer reported a problem.

Those conditions are changing quickly.

The challenge now is to remove the constraints while preserving what those assumptions were originally designed to provide: accountability, continuity, trust, and resolution.

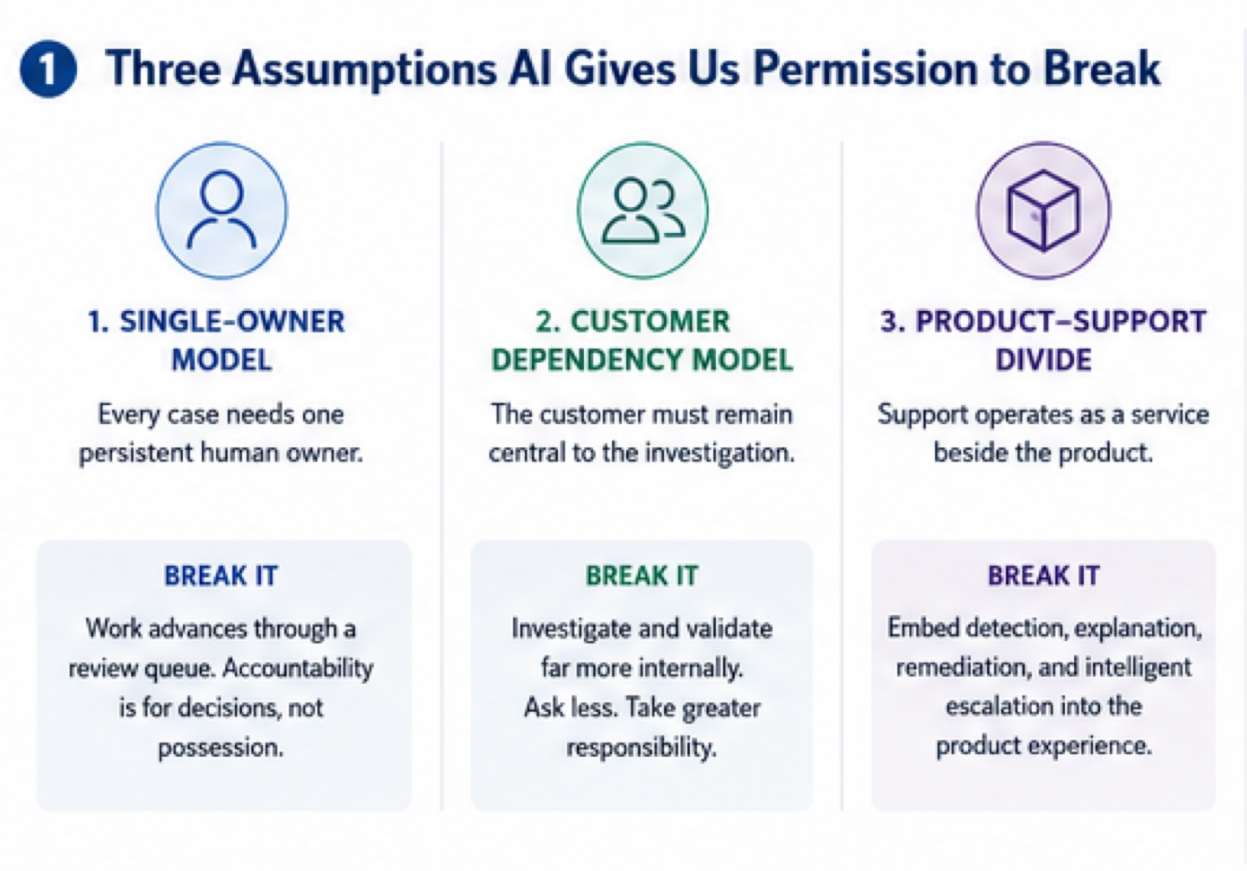


Figure 1. Three legacy assumptions that AI-enabled support can now challenge.

Assumption One: Break the Single-Owner Model

Traditional support revolves around ownership.

A case is assigned to one engineer, who coordinates the investigation, communicates with the customer, pulls in specialists, and drives the issue to closure.

The model gives everyone a clear name. It also creates a single point of dependency.

Cases wait when the owner is unavailable, handling another escalation, working in a different time zone, or searching for deeper expertise. A case can have an owner and still sit untouched for hours.

The assumption that every standard case needs one person carrying it from beginning to end should be broken.

Work advances through a review queue. AI gathers evidence, performs analysis, proposes the next step, and highlights uncertainty. Support engineers review, correct, approve, or escalate.

They take responsibility for the decision in front of them without necessarily taking possession of the full case.

That changes the unit of accountability.

The questions become more precise:

- Who confirmed the diagnosis?
- Who approved the next action?
- Who accepted the risk?
- Who decided that human escalation was required?
- Who verified the outcome?

The case can move whenever the right expertise becomes available. Continuity comes from the record of evidence, reasoning, decisions, and customer commitments.

This is a stronger model than assigning one person and hoping the work keeps moving.

Most of a case's lifecycle is often spent waiting rather than being actively worked. The exact share varies, but every support leader recognizes the pattern.

- The case waits for a specialist.
- It waits for a shift change.
- It waits for a response.
- It waits for the owner to return.

AI gives support organizations the ability to break work into smaller decisions, prepare those decisions in advance, and route them to the right person at the right moment.

Intelligent swarming and prepare-and-decide models point in this direction. The next step is to question whether persistent ownership should remain the organizing principle at all.

Removing a permanent case owner does not remove accountability. It demands sharper accountability at every consequential decision.

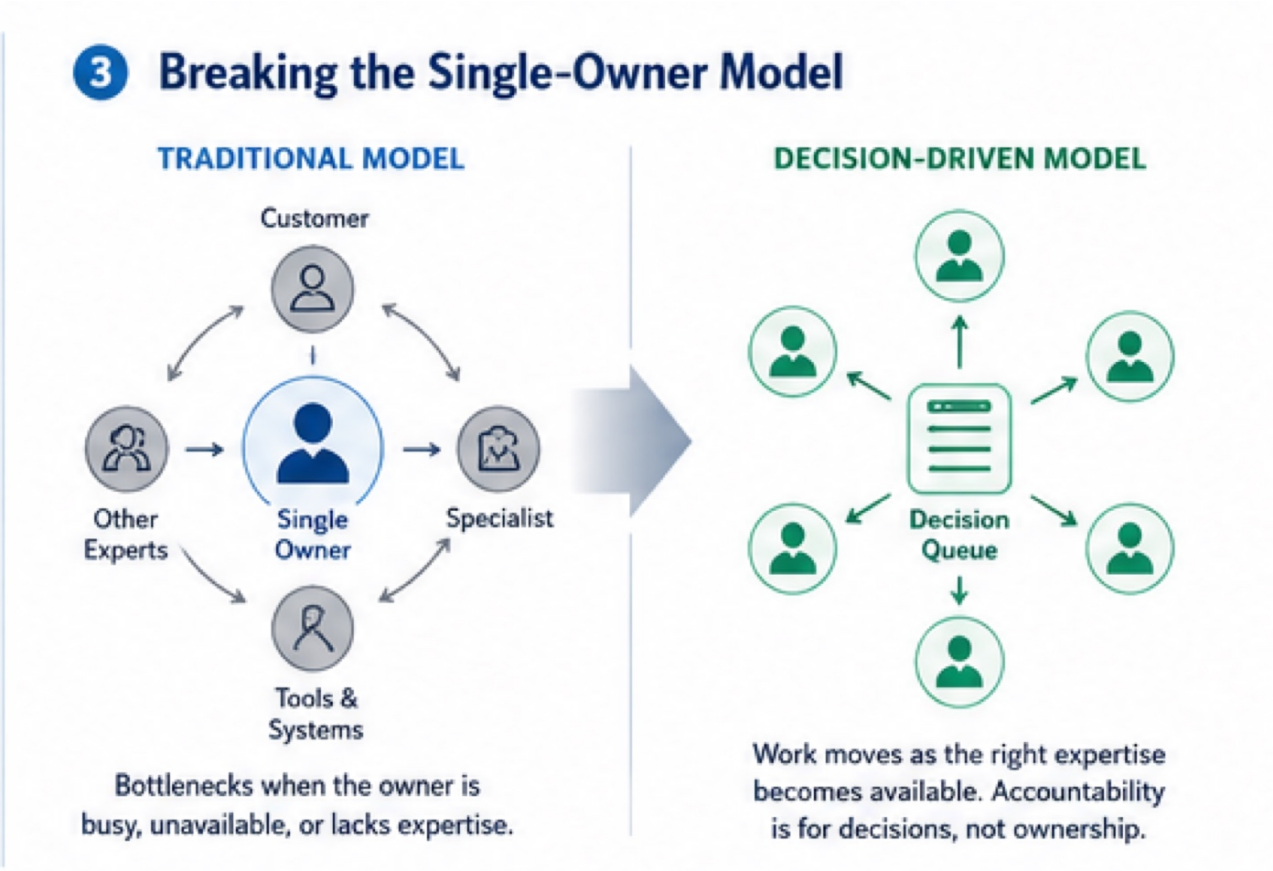


Figure 2. Decision queues allow work to move to available expertise while preserving accountability.

Assumption Two: Break the Customer Dependency Model

The second assumption is that technical investigation requires repeated work from the customer.

Support asks for logs. The customer gathers them.

Support requests configuration details. The customer searches for them.

An engineer proposes a test. The customer runs it and reports back.

Then the next round begins.

Each request may be reasonable. Together, they create a process filled with delay, customer effort, and lost context.

Customer wait states can easily consume half of a case's lifecycle. The exact share varies, but the constraint is undeniable.

Resolution will remain slow as long as the customer serves as a recurring dependency in the investigation.

Customers should provide business context, explain the impact, authorize access, and participate in decisions involving risk or operations.

They should not have to function as an extension of the support engineering team.

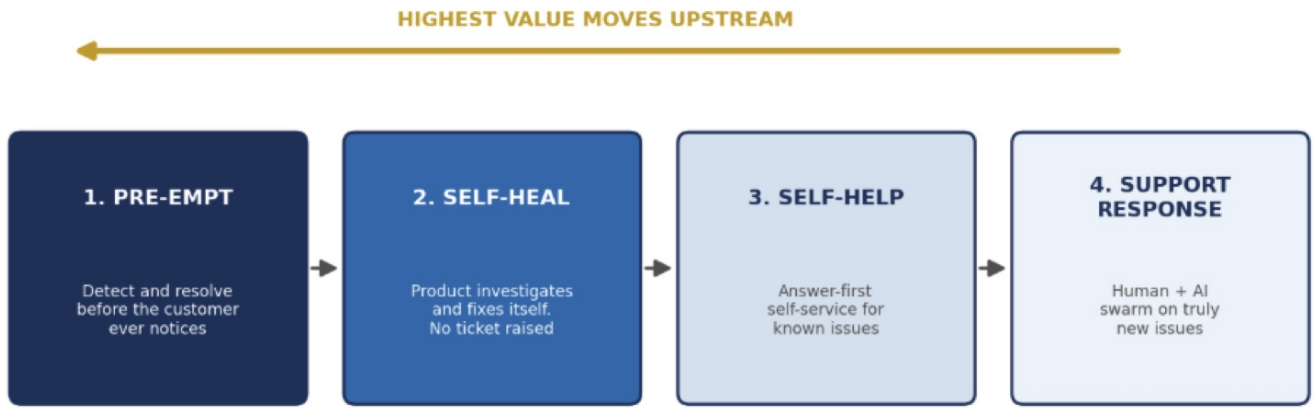
A redesigned support model carries out far more of the investigation internally.

Telemetry, observability, secure diagnostic environments, reproduction systems, and AI-assisted analysis allow the provider to gather evidence without repeatedly asking the customer to perform another task.

The workflow shifts from asking and waiting to observing, reproducing, validating, and acting.

BCG: the service value chain inverts

BCG reframes support as a four-stage value chain: pre-empt, self-heal, self-help, and support response. Most AI deployments start at the response end with deflection, summarization, and agent assist. The highest value sits at the front: preventing issues and self-healing them before a ticket exists.



Most deployments start at stage 4. The leaders are investing at stages 1 and 2.

It can start with automated evidence collection, internal reproduction, AI-prepared diagnosis, and corrective actions routed for human approval.

Each customer round trip removed from the process reduces delay, effort, ambiguity, and the risk of losing context.

This also calls for a different set of measures.

Additional measures for an AI-enabled support model

- Customer-dependent wait time
- Number of diagnostic requests made to the customer
- Percentage of diagnostic evidence gathered automatically
- Time to reproduce the issue
- Number of investigative round trips

- Percentage of cases resolved without customer-operated diagnostics

A modern support model should ask less of the customer while taking greater responsibility for understanding and resolving the issue.

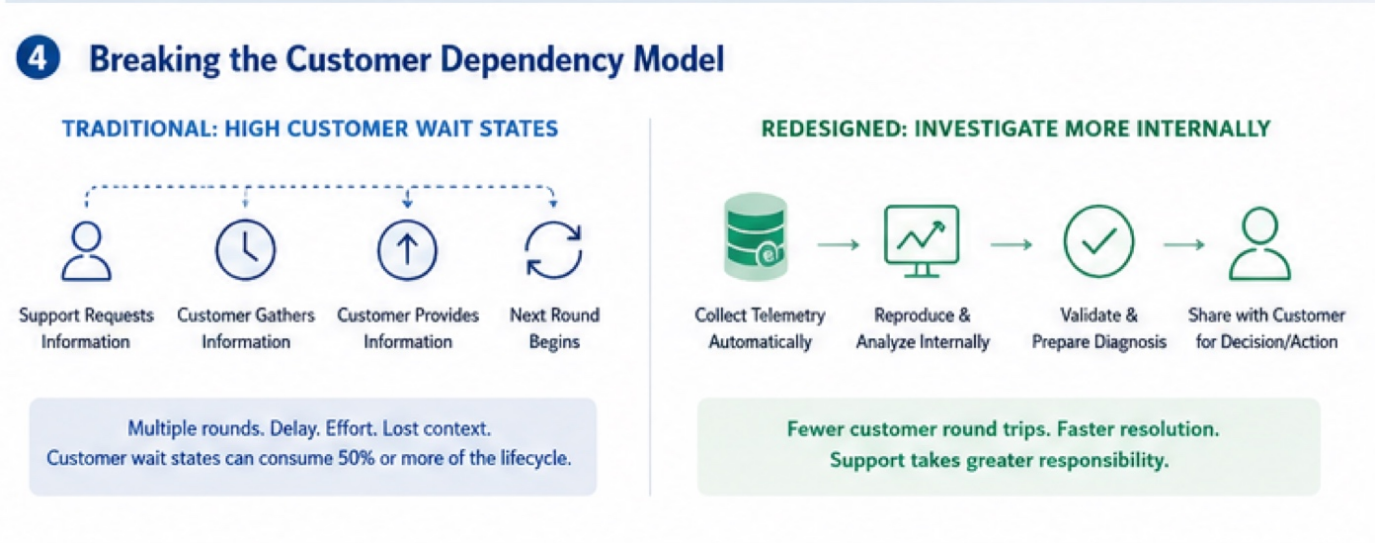


Figure 3. Internal investigation reduces customer wait states and diagnostic round trips.

Assumption Three: Break the Product-Support Divide

The third assumption is structural.

Product builds and ships the technology. Support steps in after the customer encounters a problem.

That separation has shaped organizational boundaries, budgets, platforms, and incentives for decades.

AI gives companies the ability to redesign it.

Avi described an agent owned by Product that monitors the environment, detects abnormal behavior, interprets what it means for the customer, and recommends or implements a response.

Human support enters when the issue involves ambiguity, novelty, risk, coordination, or a strained customer relationship. By then, the investigation should already be well underway.

The support experience no longer has to begin when the customer notices a problem and opens a ticket.

It can begin when the product detects that availability, performance, configuration, or a customer outcome is at risk.

- Detection becomes part of the product.
- Diagnosis becomes part of the product.
- Guidance and safe remediation become part of the product.

- Human escalation remains available when judgment and accountability are required.

The direction across the industry is consistent: greater value will come from redesigning support around prevention, product intelligence, and new operating models rather than layering AI onto existing workflows.

For support, that redesign reaches into product architecture.

Support intelligence should shape:

- Telemetry and observability
- Anomaly detection
- In-product guidance
- Configuration validation
- Automated remediation
- Release quality
- Customer health
- Engineering priorities
- Product documentation

Support teams see how products behave in real customer environments. They know where configurations fail, where upgrades introduce risk, where documentation breaks down, and where customers struggle to achieve the intended outcome.

That intelligence should flow back into the product continuously.

Support leaders should have a direct role in designing product resilience, not simply managing the consequences of product failure.

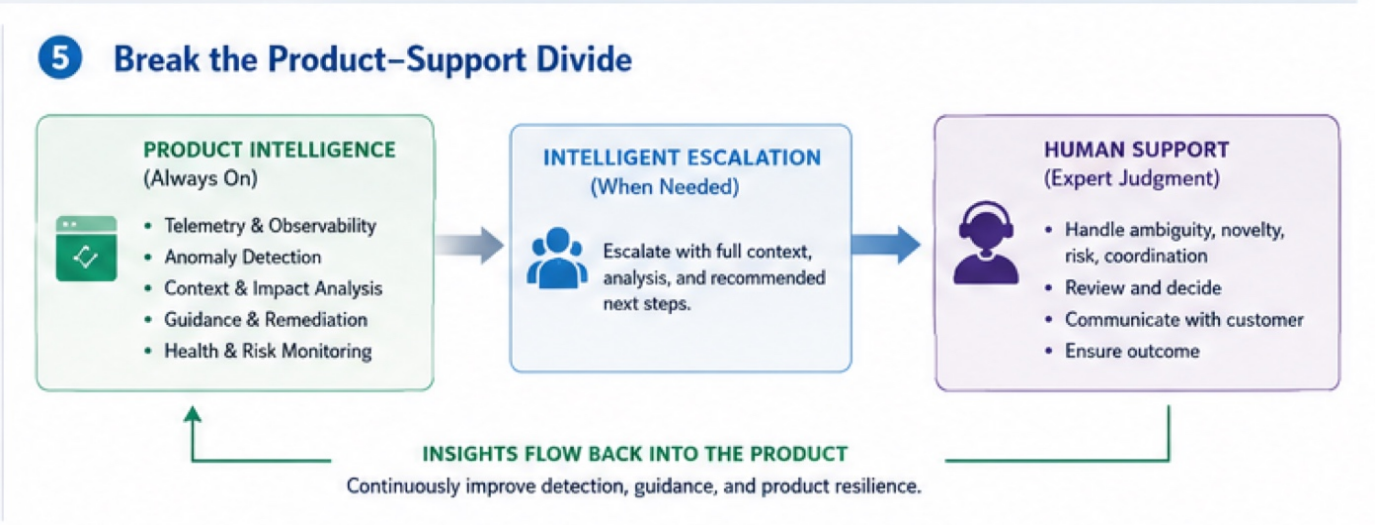


Figure 4. Product intelligence handles detection and preparation before expert human escalation.

Move from Case Flow to Decision Flow

Breaking these assumptions leads to a different operating model.

The traditional workflow follows a familiar sequence:

Operating model Workflow

Traditional case flow: Open → Assign → Investigate → Escalate → Resolve → Close.

Emerging decision flow: Detect → Understand → Assess confidence → Decide → Verify → Learn.

Cases will continue to exist. They provide a customer record, support contractual commitments, and create an audit trail.

They no longer need to dictate how the work moves.

When the case is the primary unit, work waits for the assigned person.

When the decision is the primary unit, work advances as soon as the evidence, expertise, and authority are available.

This requires more than a support automation project.

Product, Engineering, Support, Knowledge, Data, Security, Legal, and AI governance must agree on:

- What decisions agents may make
- What actions agents may execute
- What evidence is required
- How confidence is exposed
- When human review is mandatory

- How actions can be reversed
- Who remains accountable when an outcome fails

Autonomy should follow operating clarity.

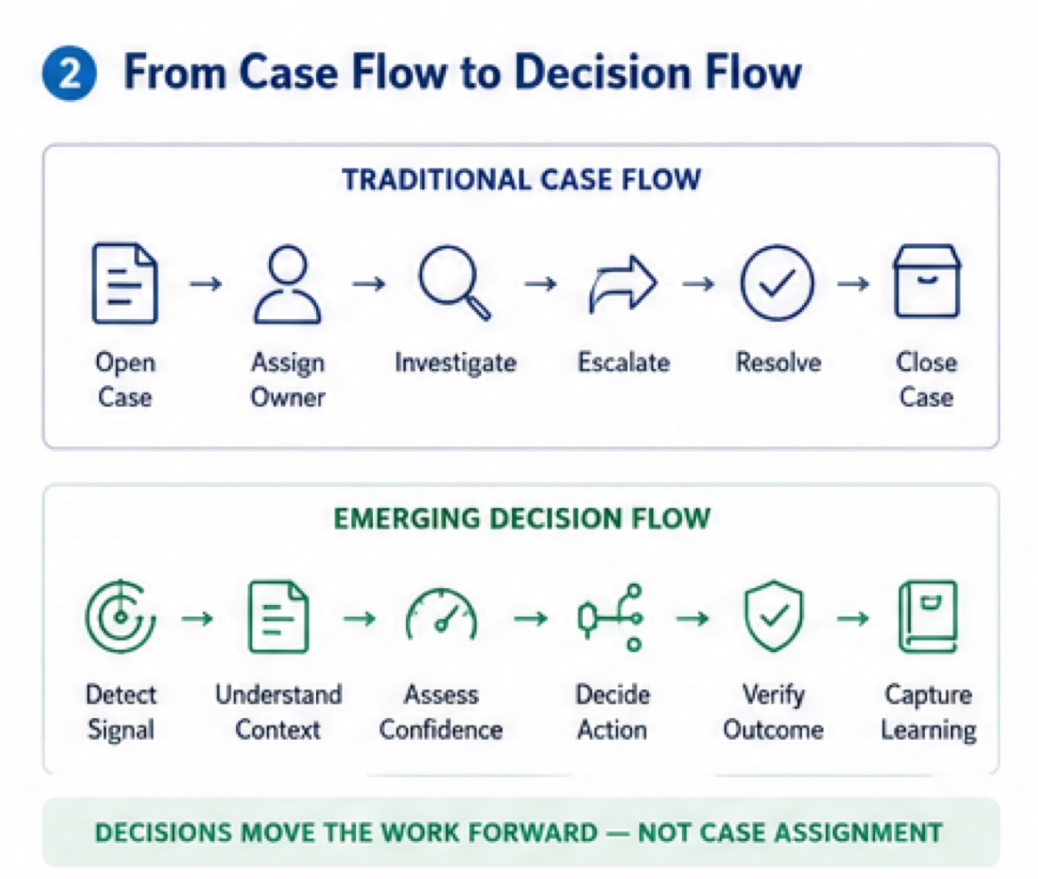


Figure 5. Support shifts from sequential case processing to continuous decision flow.

Retire Metrics, Do Not Just Add Them

Most support organizations respond to AI by adding new measures to an existing scorecard. The old metrics stay in place, the new ones sit beside them, and the operating model never actually changes.

That approach fails for a simple reason. The traditional measures were built to optimize the assumptions this article argues we should break. Average handle time rewards short interactions, not fewer of them. Cases closed per engineer reinforces single ownership. First response time measures acknowledgment rather than progress. A team can improve all three while customers wait just as long for resolution.

Measures create behavior. If we keep rewarding throughput, we will get a faster version of the model we inherited.

The stronger move is to pair each retiring metric with the measure that replaces it. The contrast is the point.

- Retire or demote: Average handle time. Replace with: Customer-dependent wait time.
- Retire or demote: Cases closed per engineer. Replace with: Number of diagnostic requests made to the customer.
- Retire or demote: First response time. Replace with: Percentage of diagnostic evidence gathered automatically.
- Retire or demote: Backlog and queue volume. Replace with: Time to reproduce the issue.
- Retire or demote: Escalation count. Replace with: Number of investigative round trips.
- Retire or demote: Case ownership and utilization. Replace with: Percentage of cases resolved without customer-operated diagnostics.

A few principles help make the transition durable.

- Measure waiting, not just working. Most of a case lifecycle is spent in wait states. Metrics that ignore waiting will miss the largest opportunity.
- Measure effort we removed from the customer. Every diagnostic request avoided is a real gain in speed, satisfaction, and retained context.
- Measure decision quality, not motion. Classification accuracy, reassignment frequency, escalation reversals, and confidence at the point of decision reveal whether the system is actually improving.
- Watch the trust signal. If AI improves the dashboard while the team trusts the system less, the metrics are measuring the wrong thing.

Retiring a familiar metric is harder than adding a new one. It requires explaining the change to executives who have tracked these numbers for years. That conversation is worth having. A scorecard built for the old architecture will quietly hold the new one back.

Treat Knowledge as Decision Infrastructure

None of this works without a different approach to knowledge.

AI cannot investigate, recommend, route, or act reliably when knowledge remains scattered across articles, case histories, engineering systems, product documentation, chat threads, telemetry, and individual memory.

Across SuccessLab forums and roundtables, support leaders have repeatedly reached the same conclusion.

Knowledge architecture is becoming the primary constraint on AI performance in support.

Most knowledge systems were designed for people. Humans can skim a long article, infer context, recognize an exception, and work around missing information.

AI agents need stronger discipline in the knowledge underneath them.

They need structure, context, provenance, ownership, boundaries, and clear rules for where the knowledge can be safely applied.

Knowledge must function as decision infrastructure.

The function that creates knowledge should remain accountable for its integrity.

- Support owns support knowledge.
- Product owns documentation and intended product behavior.
- Engineering owns technical evidence and system mechanics.
- Security owns security policy and boundaries.
- Legal owns contractual and regulatory interpretation.

Ownership should also come with an obligation to make trusted knowledge available across the company.

Knowledge cannot remain trapped inside one team or one platform.

An internal data fabric should expose authoritative knowledge to people, workflows, and agents while preserving access controls, provenance, and accountability.

The knowledge base becomes one representation of that knowledge.

Customers and engineers will increasingly receive a precise, context-aware answer rather than browse a long document. That answer may draw from product documentation, case history, telemetry, entitlement, policy, and customer context.

Reliable answers require governance as much as retrieval.

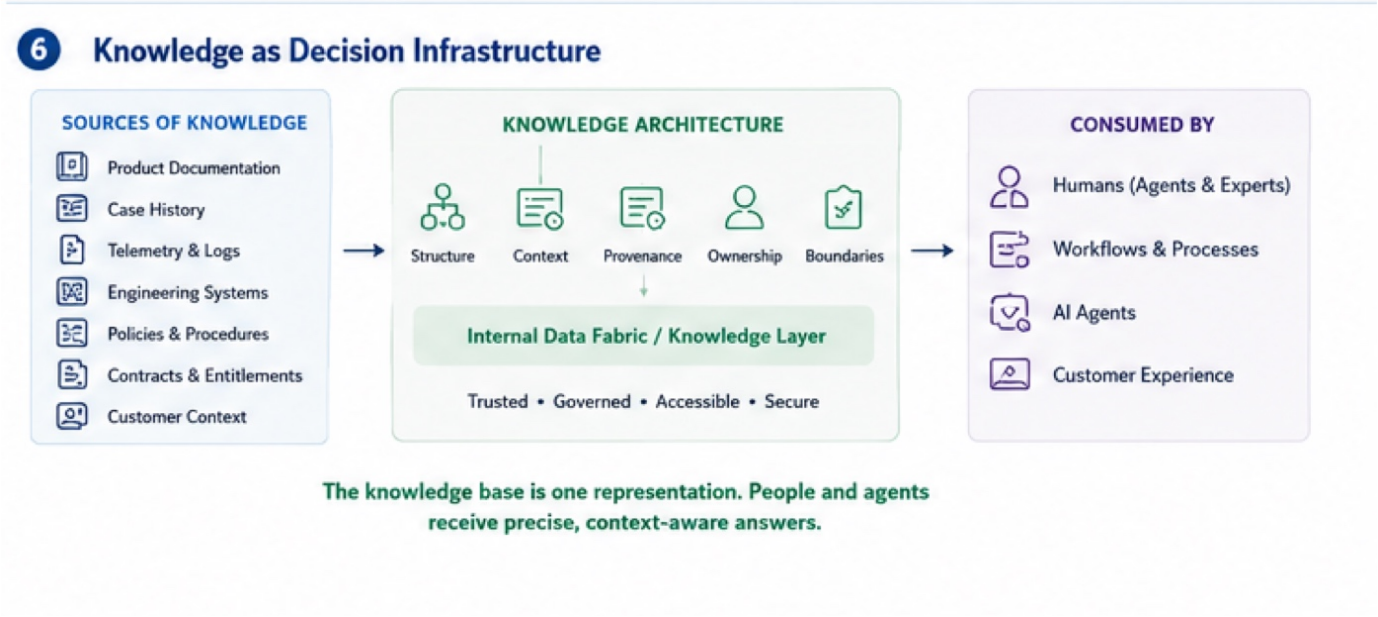


Figure 6. Knowledge becomes governed decision infrastructure for people, workflows, and AI agents.

Scaling Autonomy with Zero Trust

Enterprise AI demands ambition, but also disciplined skepticism.

Fluent answers can be wrong. Confident recommendations can rest on weak evidence. Strong demos offer little proof of how a system will perform under real production conditions.

Trust must be earned through traceability, independent validation, bounded authority, continuous monitoring, and reversibility.

That means separating creation from control.

- A diagnostic agent can propose a cause. Another system should test the evidence.
- A remediation agent can recommend an action. Policy, permissions, and risk should be checked independently.
- A resolution agent can declare the issue fixed. The outcome should still be verified.

The level of scrutiny should rise with the consequence. A low-risk answer may require light validation. A production change, security action, financial commitment, or customer promise demands much stronger controls.

Zero trust provides the discipline to scale autonomy without losing accountability. Governance must be designed into the workflow from the start.



Figure 7. Separation of duties and risk-based controls for agentic support workflows.

The Support Model We Now Need

The direction is becoming clear.

1. Break the assumption that every case requires one persistent human owner. Create explicit accountability for the decisions that move the work forward.

2. Break the assumption that the customer must remain central to the diagnostic process. Build the capability to reproduce, investigate, and validate far more internally.

3. Break the assumption that support should sit beside the product. Embed detection, explanation, remediation, and intelligent escalation into the product experience.

Knowledge provides the foundation.

Zero-trust governance makes the model scalable.

The future of technical support will be defined by how quickly companies recognize risk, apply knowledge, make sound decisions, coordinate action, and improve the product.

Cases will remain.

Human ownership will remain where continuity, judgment, and customer stewardship are essential.

Customers will remain involved where their context, consent, and priorities are required.

But these elements no longer need to define the entire architecture.

AI gives support leaders a rare opportunity to revisit assumptions that have gone largely unchallenged.

The goal should be more ambitious than automating the support model we inherited.

We should build the model customers, support teams, and products now require.

My thanks to Avi Karnon of Cribl for the thoughtful conversation and for sharing his experience and perspective.

I would welcome your perspective. Which of these assumptions are you already challenging, and where do you see the greatest opportunity or risk in rebuilding support around AI?

Join us at community.successlab.us to continue the conversation.

About the Author

Omid Razavi, Ph.D., is the founder of SuccessLab, an advisory and executive community focused on the intersection of AI transformation and enterprise customer leadership. He publishes CCO Perspectives, convenes the

SuccessLab Executive Forums and Roundtables, and advises customer, revenue, services, and support leaders across the enterprise technology industry.

More perspectives and conversations: successlab.us