EasyPost ebook

LLMs in Shipping

What Teams Are Already Doing (With or Without Permission)

easypost

If you lead a shipping, ecommerce, or operations team today, LLMs are already part of how work gets done inside your organization.

Whether you approved that or not.

Recent research shows that roughly **90% of workers report using AI tools in their day-to-day work**, often through personal or unofficial accounts, while **only about 40% of companies have formal, approved AI programs**. In practice, that means most organizations already have real AI usage happening faster than policy, governance, or strategy.

Shipping rarely fails in clean, predictable ways. Problems show up as partial information, missing scans, unclear root causes, and customers asking for answers before systems can provide them. Leaders are asked for clarity long before certainty exists.

Those are the moments where teams reach for tools that help them think clearly and respond carefully. LLMs are showing up there first, not because they're impressive, but because they help people work through uncertainty.

What LLMs actually change

At their core, large language models are pattern-recognition systems trained on language. You give them information, describe a situation, or ask a question, and they generate a response based on structure and probability.

That explanation isn't exciting, but it matters.

LLMs aren't databases. They don't replace systems of record, and they don't independently reason their way to truth. What they do well is help humans organize information, explain what might be happening, and decide what to look at next.

That's why they feel different from traditional shipping software.

Most logistics tools assume you already know what you're looking for. You choose filters, run reports, and interpret results on your own. LLMs flip that interaction. Instead of navigating menus, you describe the situation itself and ask questions in plain language. For work that starts with ambiguity, that difference is meaningful.

Guardrails that let teams move faster

The biggest risk with LLMs in shipping isn't experimentation. It's experimentation happening quietly because no one has set expectations. Clear guardrails remove that friction.

You don't need a complex policy to start. One simple rule covers most situations:

If you wouldn't paste it into Slack or email, don't paste it into an LLM.

That guideline is easy to remember, easy to explain, and removes a lot of unnecessary second-guessing. Teams understand it immediately, and it doesn't slow work down.

From there, boundaries tend to be intuitive. Most useful LLM workflows rely on redacted scenarios, aggregated metrics, internal process documents, and summaries teams already share in meetings. Where things cross the line is usually obvious: customer names, addresses, credentials, contracts, or unresolved cases that could identify a real person or shipment.

Prompting for real work

Once teams know what's safe to use, the next challenge is getting useful output. Most frustration with LLMs comes from vague prompts. A general question produces a general answer, often delivered with more confidence than it deserves.

Teams see better results when they slow down just enough to set up the problem clearly. They explain the situation, the role the model should play, and what kind of response would actually help.

I've found that one of the best ways to approach prompting is using the framework R.O.C.K.S:

R **Role** - Tell the model who it's supposed to be. An operations analyst. A CX lead. A trainer. The role shapes how it approaches the situation and what it prioritizes.

O **Output** - Say what you want back. A ranked list. A short explanation. An action plan. When the output isn't specified, the model has to guess, and guessing is where answers start to drift.

C **Context and constraints** - Share the details that actually change the answer. Service level, customer tier, timing, internal rules, or operational realities all matter more than generic background.

K **Know what's missing** - Tell it to ask clarifying questions if the information isn't sufficient. This keeps the output grounded instead of speculative.

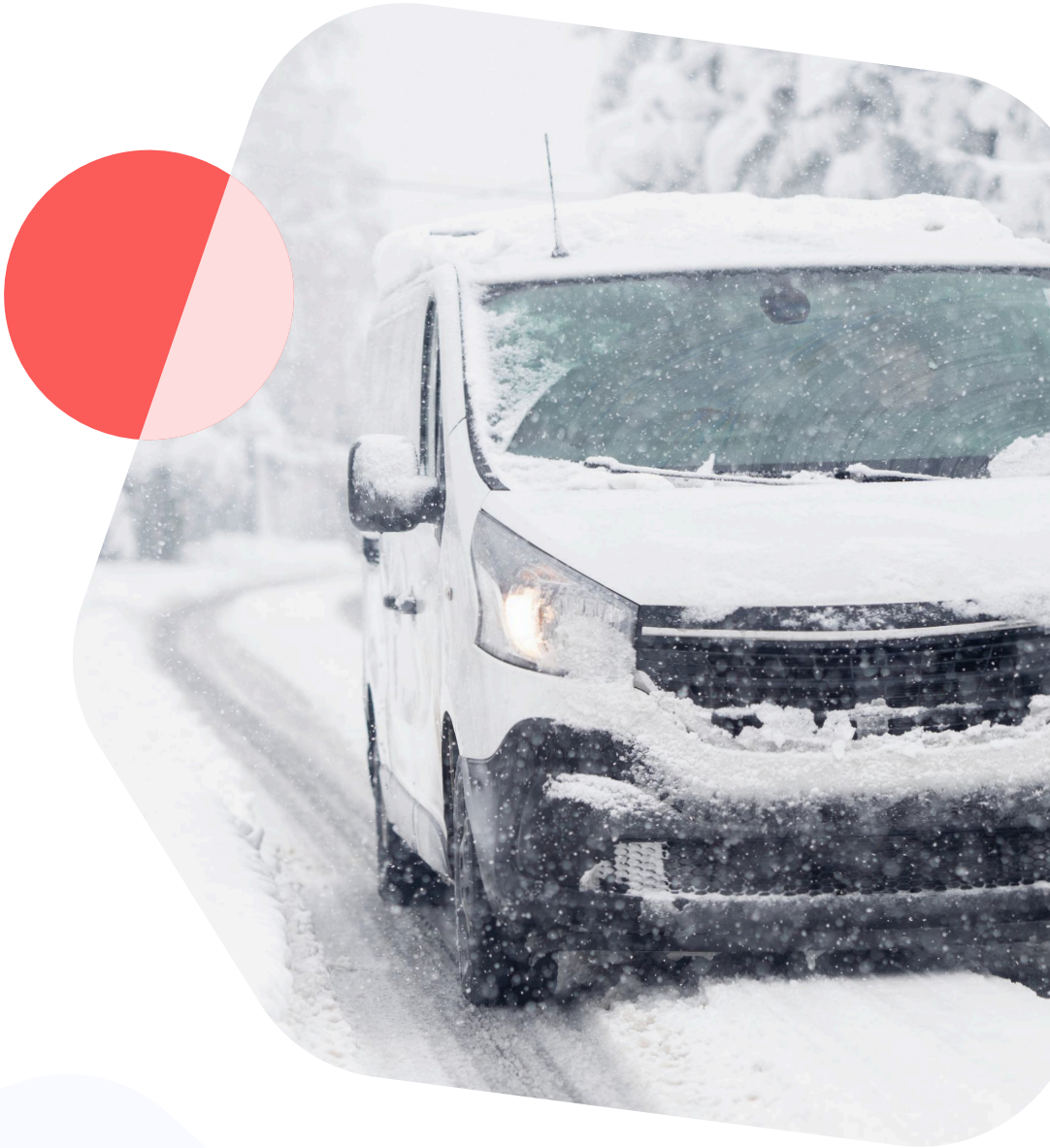
S **Show assumptions and uncertainty** - Ask it to explain what it's assuming and how confident it is. That step alone changes how teams read and use the response.

When context is clear, LLMs stop sounding generic and start supporting real decisions.

How to use the workflows

The sections that follow are intentionally practical. Each workflow reflects a real situation shipping teams deal with today. You don't need to read them in order, and you don't need all of them. Start with the one that matches where your team feels the most pressure.

The goal isn't to adopt a new process. It's to show how teams are already using LLMs to think more clearly when the work gets messy. That's where this technology proves useful.



Workflow 1: Ops Triage When a Shipment Goes Late

This workflow usually begins with a familiar escalation.

A shipment was supposed to arrive today (but didn't!) and the customer is already upset. CX needs an answer, and leadership wants to know what happened before the carrier does.

You open the system and confirm what you already suspected. The shipment is late. Some scans exist. Others don't. What's missing isn't data so much as a clear explanation and a confident next step.

This is where teams often lose time bouncing between dashboards, Slack threads, and carrier portals, trying to assemble a story under pressure.

It's also where many teams now turn to an LLM. Not to get "the answer," but to slow the moment down just enough to think clearly before responding.

Step 1: Pull together what you already know

Before writing a prompt, gather the facts you already have. It doesn't have to be perfect. You're looking for three things:

- The context of the shipment
- The route it's traveling
- The tracking story so far

For this example, the details below represent sample data types only. In real situations, teams use whatever information they already have access to, as long as it's appropriate to share and properly redacted.

Typical data sources

- TMS or order management system
- Carrier tracking page
- Customer support ticket

(You don't need everything.
You need enough to explain the situation.)

Step 2: Start with the prompt most people use

Under pressure, this is where teams usually begin:

> "This two-day shipment to Chicago is late.
> Can you tell me what happened and what I should tell the customer?"

It's understandable. It's also why the output often feels generic or overconfident.

Step 3: Flesh it out using R.O.C.K.S.

Instead of starting over, take that same prompt and layer in structure.

R **Role**

You are a shipping operations analyst supporting a customer experience team.

O **Output**

Identify the top two or three most likely causes of the delay and rank them by likelihood.
Separate internal operational actions from customer-safe communication.

C **Context and constraints**

This is a two-day replacement shipment for a top-tier customer.
The original order arrived damaged.
The promised delivery date is today, and the customer has already contacted support.
Do not promise a delivery date and do not assign blame.

K **Know what's missing**

If information is missing that would materially change the analysis, call it out explicitly.

S **Show assumptions and uncertainty**

List any assumptions you made and how confident you are in this analysis.

Now paste in the shipment details and tracking events you gathered earlier and run the prompt.

Step 4: Sanity-check the output

Good output doesn't jump to a single explanation. It ranks possibilities, references scan gaps instead of inventing certainty, and clearly separates what ops should do internally from what CX can safely say to the customer.

If the response sounds too confident given the data, that's a signal to slow down.

Step 5: Turn thinking into action

Once the situation is clearer, teams often follow up with a second prompt:

> "Turn this into a short action plan with owners for the carrier, warehouse, and CX."

This is where explanation turns into coordination without overreacting.

Step 6: Do the trust check

Before acting, ask:

> "How confident are you in this analysis, and what assumptions are driving it?"

This keeps polished answers from being mistaken for facts.

Workflow 2: WISMO and Customer Communication

This workflow usually kicks in right after ops finishes triage.
The shipment is late. There's no confirmed delivery date yet. The customer has already reached out. CX needs to respond, even though nothing meaningful has changed in the system.

This is where things often go wrong.

Under pressure, teams either overpromise, sound vague, or fall back on copy that feels cold and generic. None of those help. Teams use LLMs here for a very specific job: drafting calm, accurate communication when certainty is low.

Step 1: Get clear on what's true (and what isn't)

Before writing anything, teams align on a few facts. Not to solve the delay, just to avoid saying something they can't walk back. For this workflow, the inputs are usually very lightweight.

You're confirming:

- The shipment is delayed
- There's no confirmed delivery date yet
- The customer is already frustrated
- The carrier hasn't provided new information

For this example, the inputs below are illustrative. In real situations, teams use whatever high-level information is appropriate and properly redacted.

Typical data sources

- CX ticket or email
- TMS shipment status
- Ops notes from the triage step

Step 2: Avoid the kind of prompt that most people use

Most teams begin with something like:

> "You are a customer support specialist. Write an email explaining a delivery delay."

The result usually sounds polite and professional. It also tends to promise too much, blame the carrier, or say very little at all. That's not what you want.

Step 3: Flesh it out using R.O.C.K.S.

Instead of starting over, teams layer structure onto the prompt.

R **Role**

You are a customer support specialist for an ecommerce brand.

O **Output**

Write a short email response to a customer about a delayed shipment.
Limit it to two short paragraphs.

C **Context and constraints**

The shipment is delayed with no confirmed delivery date.
The customer has already contacted support and is frustrated.
The carrier has not provided new information.
Do not promise a delivery date.
Do not blame the carrier.

K **Know what's missing**

If the message requires information that isn't available yet, acknowledge that clearly.

S **Show assumptions and uncertainty**

Avoid definitive language. Use cautious, accurate phrasing.

At this point, teams run the prompt.

Step 4: Review the message for risk, not tone

The first read-through isn't about whether it sounds "nice." It's about whether it accidentally commits the team to something. Teams scan for:

- Implied delivery promises
- Language that shifts blame
- Vague reassurance that could be misread
- Anything that would be awkward to quote later

If something feels too confident, they revise.

Step 5: Adapt it to the channel

Once the email is solid, teams often reuse the same logic for other channels. A common follow-up prompt is:

> "Rewrite this message as an SMS under 320 characters."

Step 6: Do the trust check

Before sending, teams ask one final question:

> "Does this message promise anything we can't guarantee?"

If the answer is no, it's ready to go.

Workflow 3: Talking to Your Shipping Data

This workflow usually starts with a leadership question.

"Why are shipping costs up again?"
"Are we overusing premium services?"
"Is this a carrier problem or a process problem?"

The data exists. It's just spread across dashboards, spreadsheets, and decks. Pulling it all together takes time, and the answer usually isn't obvious anyway.

Teams use LLMs here not to replace reporting, but to **make sense of what the reports are already saying.**

Step 1: Pull a simple summary, not raw data

Before prompting, teams step back and summarize what they already know. They don't paste exports or row-level detail. They pull a short, high-level snapshot that reflects recent performance.

For this example, the data below is illustrative. In real situations, teams use aggregated or historical summaries that are appropriate to share.

A typical summary might look like:

- 18% of shipments used two-day service over the last 90 days
- 11% of two-day shipments arrived early
- 6% of two-day shipments arrived late
- Average ground delivery time: 3.4 days
- Claims filed: 420
- Claims reimbursed: 260

Typical data sources

- Monthly ops review decks
- Finance summaries
- TMS reporting dashboards
- CSV exports with sensitive fields removed

Step 2: Start with the question leadership is actually asking

Most teams jump straight to optimization. That's usually too early. A better starting point is interpretation.

A simple prompt might be:

> "Based on the summary data below, what stands out or looks unusual?"

That keeps the model focused on patterns instead of prescriptions.

Step 3: Flesh it out using R.O.C.K.S.

Teams then layer in structure so the output is useful, not speculative.

R **Role**

You are a logistics analyst supporting a shipping leadership team.

O **Output**

Identify potential cost inefficiencies or operational issues suggested by the data.
Call out patterns worth investigating further.

C **Context and constraints**

The data reflects the last 90 days of parcel shipping activity.
This is a directional review, not a final recommendation.
Do not assume changes should be made yet.

K **Know what's missing**

If the data is insufficient to draw conclusions, explain what additional data would help.

S **Show assumptions and uncertainty**

Highlight where conclusions are tentative or dependent on missing context.

Teams then paste in the summary and run the prompt.

Step 4: Read the output as a map, not an answer

Good output here doesn't tell you what to do. It tells you where to look.

Teams watch for things like:

- Premium services being used even when ground is performing well
- Early deliveries that suggest service levels may be misaligned
- Gaps between claims filed and claims reimbursed

Step 5: Ask what to investigate next

Once patterns are visible, teams often follow up with:

> "What are the top three questions we should answer before taking action?"

This is where the model helps leadership avoid jumping to conclusions too quickly.

Step 6: Do the trust check

Before anything is shared upward, teams ask:

> "What assumptions are driving this analysis, and where could it be misleading?"

That step keeps directional insight from being mistaken for fact.

Workflow 4: SOPs, Training, and Tribal Knowledge

This workflow usually starts with a familiar frustration.

The process exists. Somewhere. Parts of it are documented. Parts live in someone's head. New hires ask the same questions over and over, and experienced team members become the default help desk. Nothing is broken, exactly. It's just inefficient.

Teams use LLMs here to **make existing knowledge usable**, not to invent new rules.

Step 1: Gather what already exists

Before prompting, teams pull together whatever materials they already have. This is not about creating perfect documentation. It's about working with what's real.

For this example, the inputs below are illustrative. In practice, teams use whatever internal materials are already approved to share.

Common inputs include:

- Existing SOPs or runbooks
- Internal wiki pages
- Training decks
- Notes from senior team members
- Slack or email explanations that get reused frequently

Typical data sources

- Internal knowledge base
- Shared drives
- Onboarding materials
- Ops leadership notes

You don't need everything. One or two solid documents is usually enough to start.

Step 2: Start with the prompt most teams try

A common first attempt looks like this:

> "Summarize this SOP."

The output is usually accurate and also not very helpful. It's shorter, but it doesn't necessarily make the work easier.

Step 3: Flesh it out using R.O.C.K.S.

Teams get better results when they ask for something more specific.

R **Role**

You are an operations trainer onboarding a new team member.

O **Output**

Turn this SOP into a clear, step-by-step checklist.
Flag any steps that are decision-based rather than mechanical.

C **Context and constraints**

This process is used during high-volume periods.
The audience is a new hire with basic shipping knowledge.
Do not change the underlying policy.

K **Know what's missing**

Call out any gaps or unclear steps that would require follow-up.

S **Show assumptions and uncertainty**

Note where the SOP relies on tribal knowledge or unwritten rules.

Teams then paste in the document and run the prompt.

Step 4: Review for accuracy, not polish

The first pass isn't final. Teams review the output to make sure nothing important was lost or oversimplified. They're looking for:

- Missing exceptions
- Oversimplified decision points
- Steps that sound right but aren't operationally accurate

This review is quick, but critical.

Step 5: Repurpose instead of rewriting

Once the content is structured, teams often reuse it in different formats. Common follow-ups include:

- "Turn this checklist into a one-page quick reference."
- "Rewrite this for a seasonal temp audience."
- "Create a short training outline from this."

Step 6: Do the trust check

Before sharing widely, teams ask one final question:

> "If someone followed this exactly, would they get into trouble?"

If the answer is yes, they revise.

Workflow 5: Peak Season Planning and Scenario Prep

This workflow usually surfaces before anything has gone wrong.

Peak is coming. Volumes will rise, capacity will tighten, and something will break, even if it's not obvious yet. Leaders don't need perfect predictions at this stage. They need fewer surprises once pressure hits.

Teams use LLMs here to think through realistic scenarios before they happen, not to guess the future or replace planning work that already exists.

Step 1: Ground the discussion in what you already know

Before prompting, teams start with reality, not hypotheticals. They pull together what's already understood about the business heading into peak.

For this example, the inputs below are illustrative. In practice, teams use high-level information that's already shared internally.

Common starting points include:

- Expected volume ranges or growth assumptions
- Current carrier mix and known constraints
- Service-level commitments to customers
- Internal bottlenecks from prior peak seasons
- Promotions, launches, or campaigns already planned

Typical sources include:

- Peak planning decks
- Forecast summaries
- Post-mortems from previous years
- Sales and marketing calendars

Step 2: Start with scenarios, not solutions

Many teams jump straight to contingency plans. That often skips the most important part of planning. A better place to begin is exploration.

An opening prompt might look like this:

> "What are realistic ways peak season pressure could show up for a parcel shipping operation like ours?"

This keeps the focus on possibilities instead of panic.

Step 3: Flesh it out using R.O.C.K.S.

Once the scenario is set, teams add structure so the output stays grounded.

R **Role**

You are a logistics strategy advisor supporting a shipping leadership team.

O **Output**

Identify three to five plausible peak season disruption scenarios.
For each, briefly describe the operational impact.

C **Context and constraints**

The business ships parcel at scale.
Customer expectations around delivery speed are high.
The goal is preparation, not prediction.

K **Know what's missing**

Call out any assumptions you're making about volume, carrier capacity, or customer behavior.

S **Show assumptions and uncertainty**

Note where scenarios depend on factors outside the team's control.

Teams then add their high-level inputs and run the prompt.

Step 4: Pressure-test the scenarios

Once scenarios are on the table, teams focus on what each one would actually require if it started to unfold. Common follow-up prompts include:

- "What early warning signs would tell us this scenario is emerging?"
- "What decisions would we need to make quickly if this happens?"
- "Which teams would be involved first?"

This step turns abstract risk into something operational.

Step 5: Identify no-regret moves

From there, teams look for actions that would help across multiple scenarios, not just one.

These often include:

- Clear escalation paths
- Pre-approved customer messaging
- Carrier communication plans
- Internal decision thresholds

A useful framing question here is:

> "What could we put in place now that would help regardless of which scenario plays out?"

That's where preparation becomes concrete.

Step 6: Do the trust check

Before treating any of this as a plan, teams ask:

> "Which of these scenarios are we over-indexing on, and which might we be underestimating?"

That question keeps planning grounded instead of reactive.

The Real Opportunity Ahead

If there's one takeaway from these workflows, it's that LLMs are already part of how shipping teams work, whether that use is formal or not. People reach for them when the work gets messy and the answers aren't obvious.

That's not about chasing new technology. It's about needing help thinking clearly under pressure.

Across ops triage, customer communication, data analysis, training, and peak planning, the pattern is consistent. LLMs are most useful when they help teams organize what they know, surface what they don't, and communicate with more care. They don't replace systems or judgment. They support better decisions when time and certainty are in short supply.

For leaders, the question isn't whether to adopt AI. It's where ambiguity creates the most risk today and where clearer thinking would change outcomes. Everything here focused on how teams already operate. What comes next is simply extending these capabilities into the systems teams rely on every day.

LLMs Built Specifically for Shipping?

Throughout this ebook, the examples have followed a consistent pattern. Teams encounter uncertainty, pull together the information they already have, and use an LLM to help them think clearly before acting. Usually those LLMs are things like ChatGPT or Gemini.

The natural next step is not more AI.
It's AI that lives where the work already happens.

At EasyPost, this is what led us to build **Luma Advisor**, a shipping-focused LLM designed to support the kinds of decisions covered in this ebook. Instead of pasting shipment details into a general-purpose tool, teams can ask questions directly against their own shipping data and receive clear, plain-language answers in return.

The goal isn't automation for its own sake. It's the same goal you've seen throughout these workflows: helping teams understand what's happening, why it's happening, and what to do next without digging through reports or guessing under pressure.

Luma Advisor builds on the same principles you've already seen:

- Clear roles and expectations
- Real operational context
- Guardrails that protect trust
- Answers that surface assumptions instead of hiding them

It's not meant to replace systems, people, or judgment. It's meant to make judgment easier when complexity gets in the way.

This ebook focused on how teams are already using LLMs today. What comes next is simply making that work more intentional, more accurate, and more connected to the data shipping teams rely on every day.