Recipient Consent for Third-Party Deliveries

Designing a Trust & Safety experience that gives recipients control over unwanted deliveries while balancing customer experience and operational efficiency.

CASE STUDY | TRUST & SAFETY | QUICK COMMERCE

Background: Where This Began

This case study was inspired by an experience shared by a close friend.

After cutting contact with someone due to repeated toxic behavior, that person used Zepto—a popular quick-commerce platform—to send groceries to her address as a way to re-establish communication. While the order itself wasn't harmful, it took away her ability to choose whether she wanted any interaction with that person.

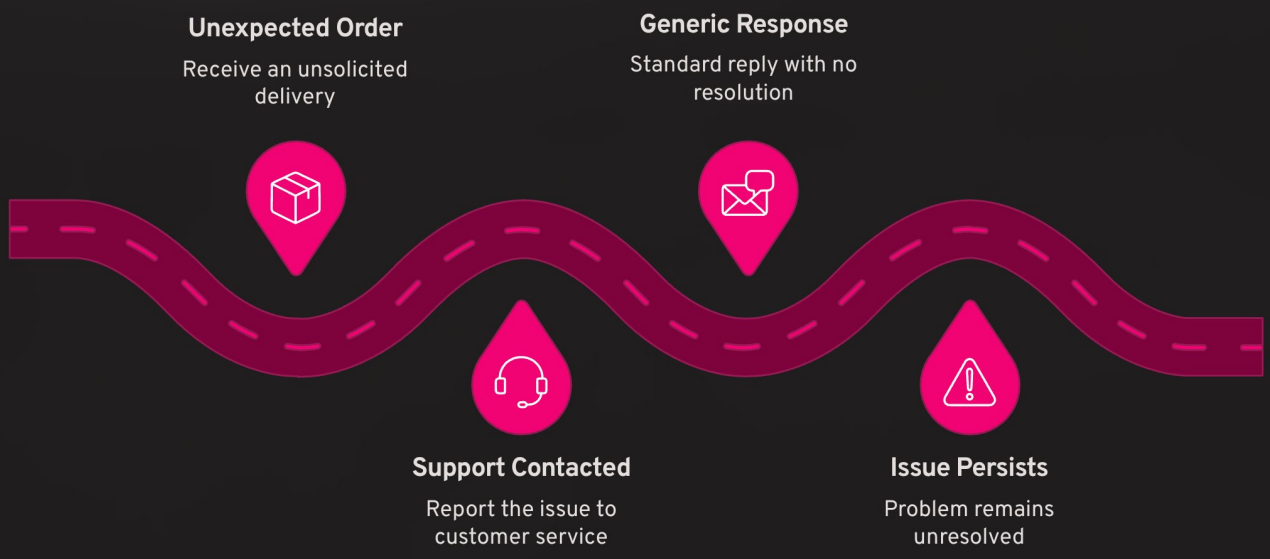
She looked for a way to cancel or reject the order before it arrived but couldn't find one. There was no option to decline the delivery, no recipient controls, and no way to prevent similar deliveries from the same sender in the future.

Watching her navigate this situation made me realize that while Zepto is designed to make ordering effortless for the sender, it offers very little control to the recipient when a delivery is unwanted, uncomfortable, or emotionally distressing.

What began as a friend's frustrating experience soon became a product design question:

Why are quick-commerce platforms built entirely around the sender's intent, while giving the recipient almost no agency?

That question became the foundation for exploring how recipient consent could be incorporated into the quick-commerce experience through a Trust & Safety lens.



The Customer Support Experience

Hoping to stop the delivery before it arrived, my friend first tried resolving the issue through Zepto's in-app support. However, the chatbot only handled a limited set of predefined queries and didn't provide an option for this situation.

With no way to speak to a support representative, she emailed Zepto explaining that the order had been placed by someone else using her name and delivery address, and requested that it be cancelled before delivery. At the time of writing this case study, no response had been received for the query raised.

This highlighted a larger product gap. The platform had no workflow for recipients facing unwanted third-party deliveries, leaving them without a clear path to seek help or take action.

That insight became the starting point for this product exploration.

What Happened

Unexpected third-party delivery placed without recipient's knowledge or consent.

What Was Tried

Contacted customer support requesting cancellation or blocking.

What Was Offered

Generic response focused on order status, no resolution for safety concern.

What Was Missing

Any recipient-facing control, block mechanism, or safety escalation path.

Problem Statement

Quick-commerce platforms are designed primarily around the sender, optimizing every step of the ordering journey. However, recipients can receive deliveries they neither requested nor want.

Situations such as harassment by former partners, stalking, prank deliveries, domestic abuse, or wrong-address orders reveal an important product gap: recipients have little to no control over unexpected third-party deliveries.

The current experience assumes every delivery is welcome. In reality, that's not always the case.



Harassment by Former Partners

Deliveries used as a pretext to re-establish contact after a relationship has ended.



Stalking & Surveillance

Confirming a target's address or presence through repeated delivery attempts.



Domestic Abuse Situations

Deliveries used as coercive control tactics by abusers with known addresses.



Prank & Unwanted Deliveries

Unsolicited deliveries sent as a joke or to cause inconvenience without consent.

> **"The real problem wasn't the inability to cancel an order — it was the absence of recipient control."**

User Persona: Arpitha

Arpitha

Working Professional, Bengaluru

Uses Zepto multiple times every week for groceries and household essentials. Tech-savvy, privacy-conscious, and accustomed to managing her life through apps.

> "I should have a say in what gets delivered to my doorstep, especially when it's being used as a way to contact me."

Goals

- Receive only deliveries she expects and has consented to
- Feel safe and in control while using delivery platforms
- Resolve issues quickly without depending on customer support
- Maintain personal boundaries without explanation or justification

Pain Points

- Cannot stop unwanted third-party deliveries through any in-app mechanism
- Forced into unwanted interactions or contact attempts via platform
- Customer support cannot prevent a delivery already in progress
- No ability to block repeat senders or flag safety concerns proactively

User Journey Map

Mapping Arpitha's emotional journey through an unwanted third-party delivery reveals exactly where the platform fails her — and where design interventions would have the highest impact.

Order Placed

Sender places order without recipient's knowledge. Arpitha has no awareness yet.

Neutral

Notification Received

Arpitha gets an app notification. Surprise and discomfort begin immediately.

Anxious

Attempts Support

Reaches out to customer support hoping for a resolution. Hopeful but uncertain.

Frustrated

No Resolution

Chatbot can't resolve the issue. Emails support, no response received. Feels a lack of control.

Defeated

Delivery Arrives

Delivery arrives at door. Forced interaction with the situation she tried to avoid.

Violated

Lack of Control

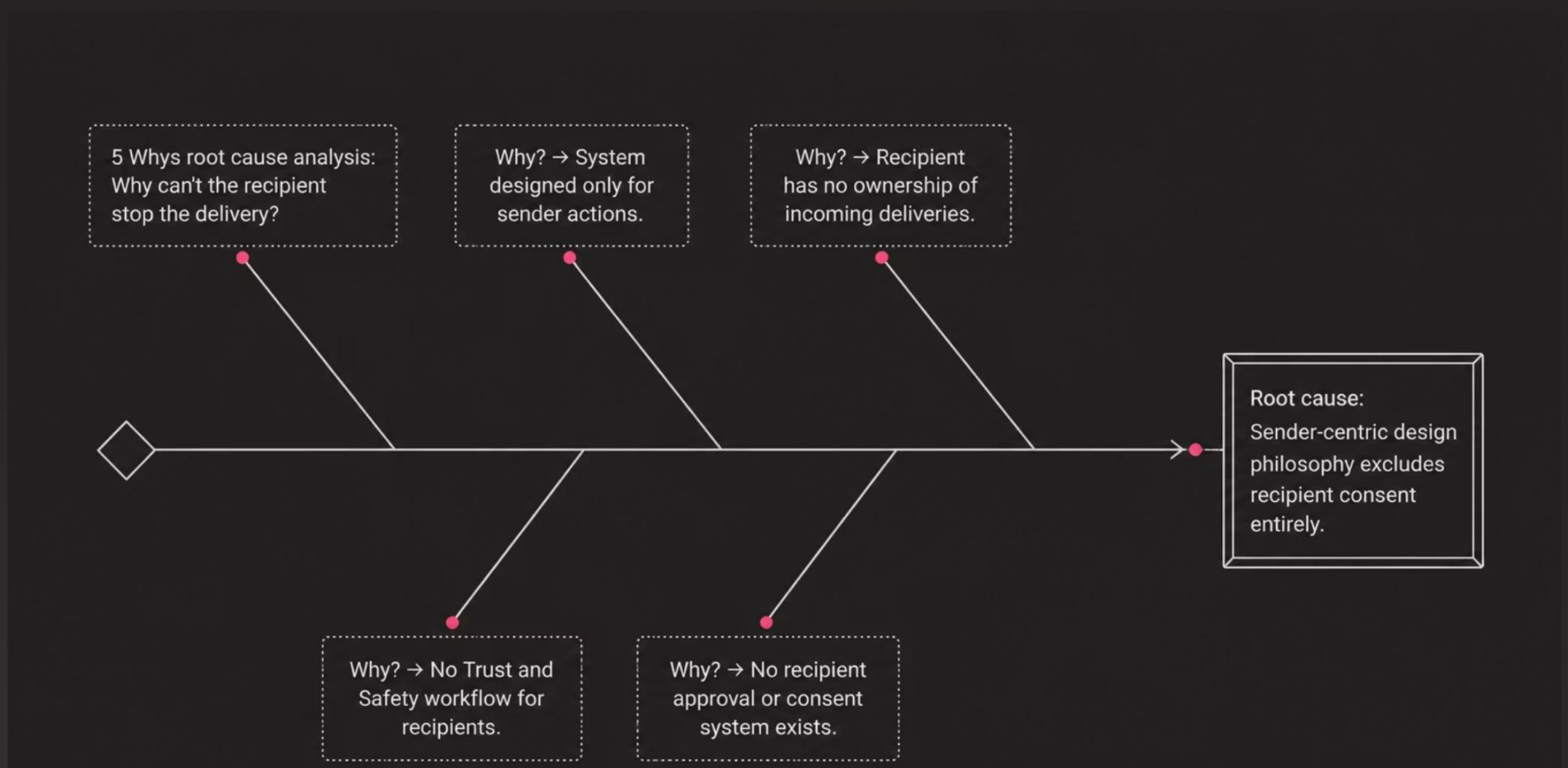
Reflects on the experience with a lasting sense of platform distrust and vulnerability.

Distrust

The emotional arc is consistently downward — from neutral surprise to a lasting sense of violation and distrust. Every touchpoint after the initial notification represents a missed opportunity for the platform to restore the recipient's sense of agency. The most critical intervention window is between notification and order fulfillment, which is precisely where the proposed solution targets.

Root Cause Analysis

Using a 5 Whys methodology, we can trace the surface-level complaint — "I can't stop this delivery" — back to its root cause in platform philosophy and system architecture.



The fishbone analysis reveals that the problem is not a missing feature — it is a missing **design philosophy**. Every layer of the system, from order flow to support tooling to notification architecture, was built assuming the sender's intent is sufficient to authorize a delivery. Recipient agency was never factored into the design equation. Solving this requires not just adding a button, but rethinking who the platform serves in a third-party delivery transaction.

Product Vision & Proposed Solution

VISION

> **Every recipient should have the ability to decide whether an unexpected third-party delivery reaches their doorstep.**

The proposed shift is simple but significant: move from Sender-Centric Commerce to Consent-Driven Deliveries. This doesn't mean every delivery requires approval—trusted contacts can bypass the flow. Instead, it gives recipients meaningful, real-time control when an unexpected or unwanted sender initiates a delivery.

Recipient Consent Flow

When a third-party order is placed, the recipient receives a notification: *"Someone has sent you groceries."* They can Accept or Decline before fulfillment begins. If declined, the order is cancelled with a neutral message to the sender.

Block Sender

After declining, the recipient is offered the option to block future deliveries from the same sender. All future attempts are automatically rejected without requiring repeated action.

Trusted Circle

Users can designate Family, Friends, and Emergency Contacts. Deliveries from trusted people skip the approval step entirely, preserving a frictionless experience for expected orders.

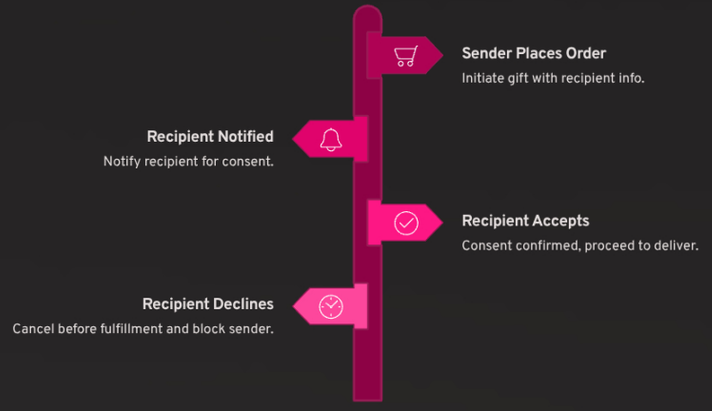
Safety Mode

If multiple unwanted attempts are detected from the same sender, the platform proactively offers options to report harassment, escalate to support, or permanently block the account.

Critically, when a recipient declines, the sender is notified only that *"Delivery could not be completed"*— a neutral message that does not explicitly reveal the rejection. This design choice is intentional: explicitly communicating rejection to a harasser can escalate danger. The platform's role is to protect the recipient, not to adjudicate between parties.

User Flow & Edge Cases

Recipient Consent Flow



Edge Case Handling

Scenario	Solution
Friend sending a birthday gift	Trusted contacts bypass approval automatically
Sender enters wrong address	Recipient declines before fulfillment begins
Order already picked up by rider	Recipient informed; refusal option offered at delivery
Repeated harassment attempts	Automatic Trust & Safety review triggered

Trade-offs & Success Metrics

Benefits vs. Challenges

Benefits	Challenges
Better user safety and wellbeing	Additional operational complexity
More recipient control and agency	Potential increase in order cancellations
Reduced support ticket dependency	Refund handling and processing overhead
Increased platform trust and loyalty	Food wastage from last-minute declines

Mitigations: Consent approval window limited to pre-fulfillment phase. Trusted Contacts reduce friction for most orders. Recipient consent only required for third-party orders, not self-orders.

Success Metrics

80%

Fewer Complaints

Reduction in recipient safety complaints

90%

Block Effectiveness

Reduction in repeat unwanted deliveries after blocking

30s

Resolution Time

Target recipient issue resolution under 30 seconds

4.5+

CSAT Score

Target customer satisfaction score out of 5

The North Star Metric for this feature is the **percentage of unwanted third-party deliveries successfully prevented before delivery**. Business-level tracking should also include: reduction in Trust & Safety support tickets, adoption rate of Trusted Contacts, improvement in recipient trust scores, and reduction in operational support costs associated with delivery disputes.

What I Learned

This case study changed how I think about product design. I initially saw the problem as a missing cancellation feature, but I soon realized the real issue was giving recipients control over unwanted interactions.

It reminded me that great products aren't built only for happy paths—they're also designed for moments when users feel vulnerable. More importantly, it reinforced that good product management is about understanding the need behind a feature request and building solutions that balance user trust, business goals, and operational constraints.

> **"The best product solutions don't just solve functional problems — they give users control when they need it the most."**