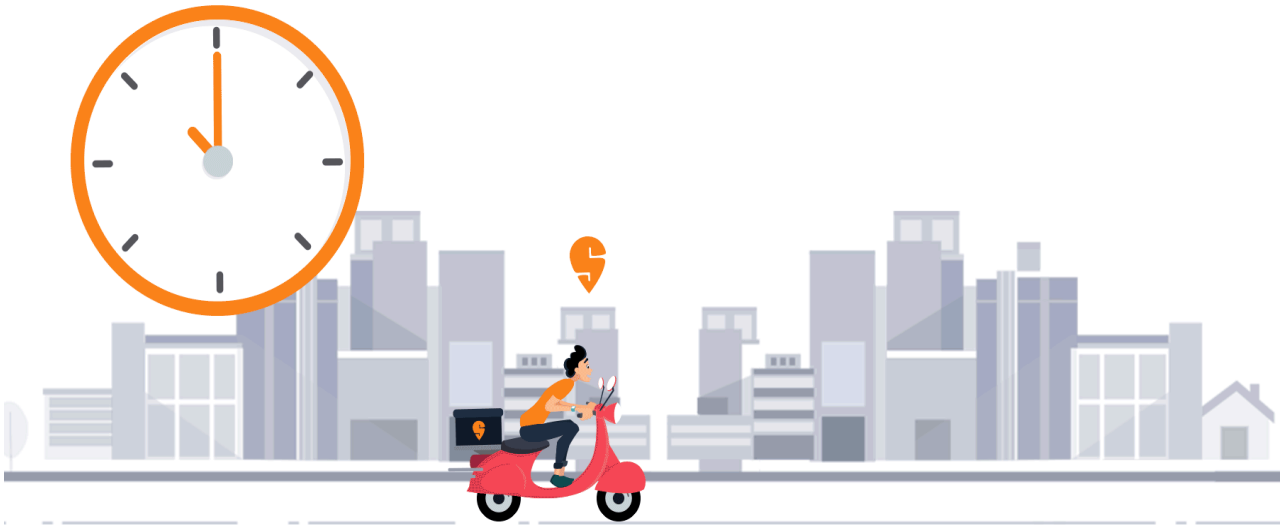
CASE STUDY



PROBLEM STATEMENT

Mapping Business Outcomes to Product Outcomes: Swiggy's Customer Support Challenge

Made By - Anjan Mahapatra (Cohort 36)

Executive Summary :

Swiggy is an on-demand hyperlocal delivery platform that primarily facilitates online food ordering and delivery services. In addition to restaurant meals, Swiggy has expanded its offerings to include grocery delivery (Swiggy Instamart), pick-and-drop services (Swiggy Genie), and subscription-based convenience (Swiggy One). It leverages a vast logistics network and a mobile-first user interface to provide seamless experiences for millions of users across India.

Swiggy, launched in 2014, is India's leading food delivery platform operating in over 500 cities. With increasing order volumes and platform usage, Swiggy has seen a proportional rise in customer support queries—many of which are repetitive and could be resolved through better product design and automation. The current challenge lies in managing these queries efficiently while maintaining high customer satisfaction and operational effectiveness.

Business Objective :

Primary Goal :

Reduce the volume of customer support queries.

Swiggy's customer support function plays a critical role in ensuring user satisfaction and platform reliability. With increasing order volumes, Swiggy is experiencing a rising influx of customer queries, many of which are repetitive and rooted in predictable friction points such as delayed deliveries, unclear refund policies, and ineffective in-app help tools.

The issue will not merely address the operational necessity but also strategic business imperative.
A reduction in support queries will translate into:

1. **Improved customer experience :**
   Faster resolutions and fewer support interactions reduce customer frustration.

2. **Operational cost savings :**
   Fewer queries mean lower reliance on human agents and the ability to scale more efficiently.

3. **High retention and loyalty :**
   Frictionless experiences foster long-term customer relationships.

4. **Increased agent productivity :**
   Streamlined support enables agents to focus on high-complexity or high-priority cases.

To achieve this, Swiggy must take a product-first approach that targets the root causes of support demand, automates resolution wherever possible, and empowers users to resolve issues independently.

Root Cause Hypothesis :

In analyzing the surge in customer support queries, we identify several recurring themes that point to deeper systemic and product-level issues.

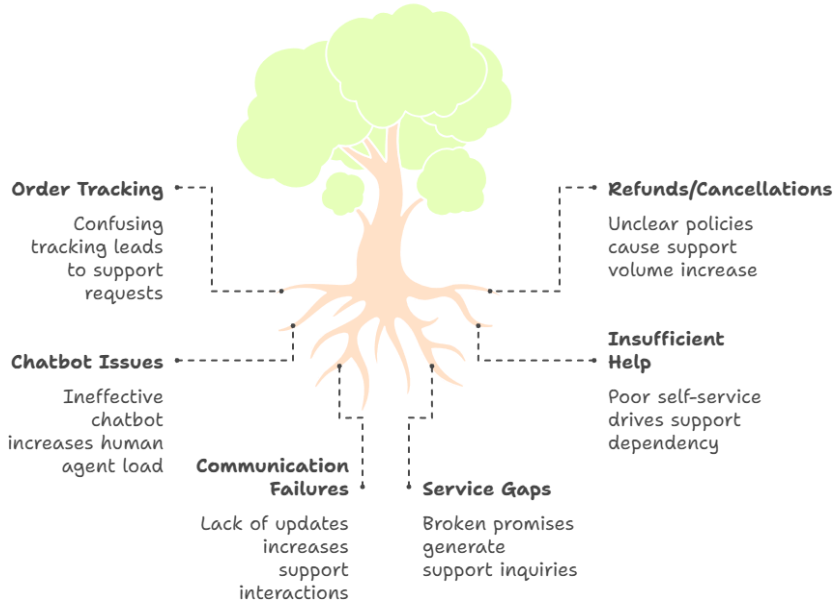


Figure : Root Cause Hypothesis

Each hypothesis below reflects a category of problems, the likely user scenarios associated with them, and their consequences for the support workload:

1) **Order Tracking Confusion and Delays:**

   Scenario: Customers are frequently unsure about the real-time status of their order, especially during peak hours, rains, or high traffic zones.

   Root Issue: Inaccurate ETAs, lack of timely updates, or sudden changes in delivery route.

   Impact: High volume of "**Where is my order?**" queries that could be prevented with better tracking transparency.

2) **Refund and Cancellation Uncertainty:**

   Scenario: Customers cancel due to issues like restaurant unavailability or delivery delays, and then wait anxiously for refunds.

   Root Issue: No clear or timely updates on refund processing status or timeline.

   Impact: Repeated refund status inquiries, low trust, and poor customer satisfaction.

3) **Chatbot Ineffectiveness and Escalation Barriers:**

Scenario: Users are first routed through a chatbot, which fails to understand intent.

Root Issue: Poor intent recognition, lack of fallback mechanisms, and rigid escalation workflows.

Impact: Increased frustration and unnecessary ticket escalations to human agents.

4) **Inadequate Self-Help or In-App Support Flows:**

Scenario: Customers struggle to find relevant help or FAQs in the app during time-sensitive situations.

Root Issue: Help center is buried in the UI, outdated content, or lacks scenario-based guidance.

Impact: Higher dependency on contacting support for basic information.

5) **Lack of Proactive Communication:**

Scenario:Unexpected delays, cancellations, or substitutions are not communicated clearly in real-time.

Root Issue: Absence of push notifications, in-app banners, or real-time SMS updates.

Impact: Confused users seek manual support, leading to preventable queries.

6) **Payments and Transaction-Related Queries:**

Scenario: Payments fail, get debited without confirmation, or are not reflected immediately.

Root Issue: Payment gateway issues, slow reconciliation, or poor UI feedback.

Impact: Disproportionate number of queries related to payment status and double charges.

7) **Service Promise Gaps (e.g., Swiggy One, Instamart, Genie):**

Scenario: Users don't receive promised benefits like priority delivery, free delivery, or special discounts.

Root Issue: Misalignment between service expectations and real-time execution.

Impact: Support queries tied to subscription dissatisfaction and confusion.

8) **Order Issues at the Restaurant Level:**

Scenario: Wrong items delivered, items missing, or restaurants unresponsive after order placement.

Root Issue: Restaurant-level operational inconsistencies or miscommunication.

Impact: Post-order complaints that often escalate to refund or compensation requests.

- **Estimating Customer Support Staffing Needs :**

Accurately estimating support staffing is crucial to ensure customer issues are resolved efficiently without overloading the operations team or incurring unnecessary costs.

Below is a detailed model to estimate the required number of customer support agents for Swiggy based on common industry benchmarks and scalable planning methodologies.

Key Assumptions:

Total Monthly Orders : ~50 million (based on industry reports and Swiggy's scale)

Support Query Rate : ~2% of total orders result in support interactions (consistent with benchmarks from similar platforms like Zomato, Uber Eats, and Doordash – source: Freshworks, Zendesk Benchmark Reports)

Total Monthly Queries : ~1 million (50M x 2%)

Productivity per Agent: 1,000 queries/month per agent (based on global customer support team productivity benchmarks and Zendesk's 2023 CX Trends Report)

Notes:

(i) 1,000 agents are needed under current conditions, assuming no automation or product improvements.

(ii) This number can vary based on support channel (chat vs. email vs. call), query complexity, and use of AI/chatbot triaging.

(iii) Platforms like Intercom and Zendesk suggest that best-in-class support operations aim to resolve at least 30–40% of queries via self-service or automation.

With Automation in Place:

If Swiggy implements effective product-led resolutions (e.g., chatbot, self-serve refund flows), and deflects 40% of queries:

Adjusted Monthly Queries = 1,000,000 x (1 - 0.4) = 600,000

Revised Agent Need = 600,000 / 1,000 = 600 agents

Thus, **automating even 40% of support load** could result in a **40% reduction in required headcount**, leading to significant operational cost savings and improved agent efficiency.

- **Product Outcome (Mapped to Root Causes) :**

To ensure targeted reduction in customer support queries, Swiggy must translate each identified root cause into actionable product outcomes.

The following matrix offers a detailed mapping between root causes, user pain points, and product outcomes with defined success metrics:

Root Cause Category	User Pain Point	Product Outcome	Key Metrics (KPIs)
Order Tracking Confusion and Delays	Unclear or incorrect ETA, no live updates	Real-time delivery tracking with dynamic ETA updates	Reduction in "Where is my order?" queries; ETA adherence rate
Refund and Cancellation Uncertainty	Lack of refund visibility, unclear status post-cancellation	Self-serve refund and cancellation flows with status tracker	% drop in refund-related queries; resolution time
Chatbot Ineffectiveness	Chatbot gives generic or irrelevant replies	AI-driven chatbot with escalation support and user context memory	Chatbot resolution rate; % queries resolved without escalation
Inadequate Self-Help UX	Users can't find relevant help during order issues	Context-aware FAQ and scenario-based guided flows	FAQ engagement rate; % queries avoided through self-help
Lack of Proactive Communication	Users unaware of delays, substitutions, or cancellations	Push notifications and proactive order alerts (in-app/SMS)	Reduction in repetitive queries; CSAT
Payments and Transaction Confusion	Payment failed or delayed reflection	Real-time payment confirmation with retry or complaint option	% drop in payment-related queries; resolution turnaround
Service Promise Mismatches	Subscriptions not delivering promised benefits	Unified dashboard showing subscription status and benefits	Subscription-related complaint frequency; NPS from subscribers
Restaurant-Level Issues	Items missing, wrong orders, unresponsive merchant	Post-order feedback and auto-compensation flows	% auto-resolved issues; complaint-to-resolution ratio

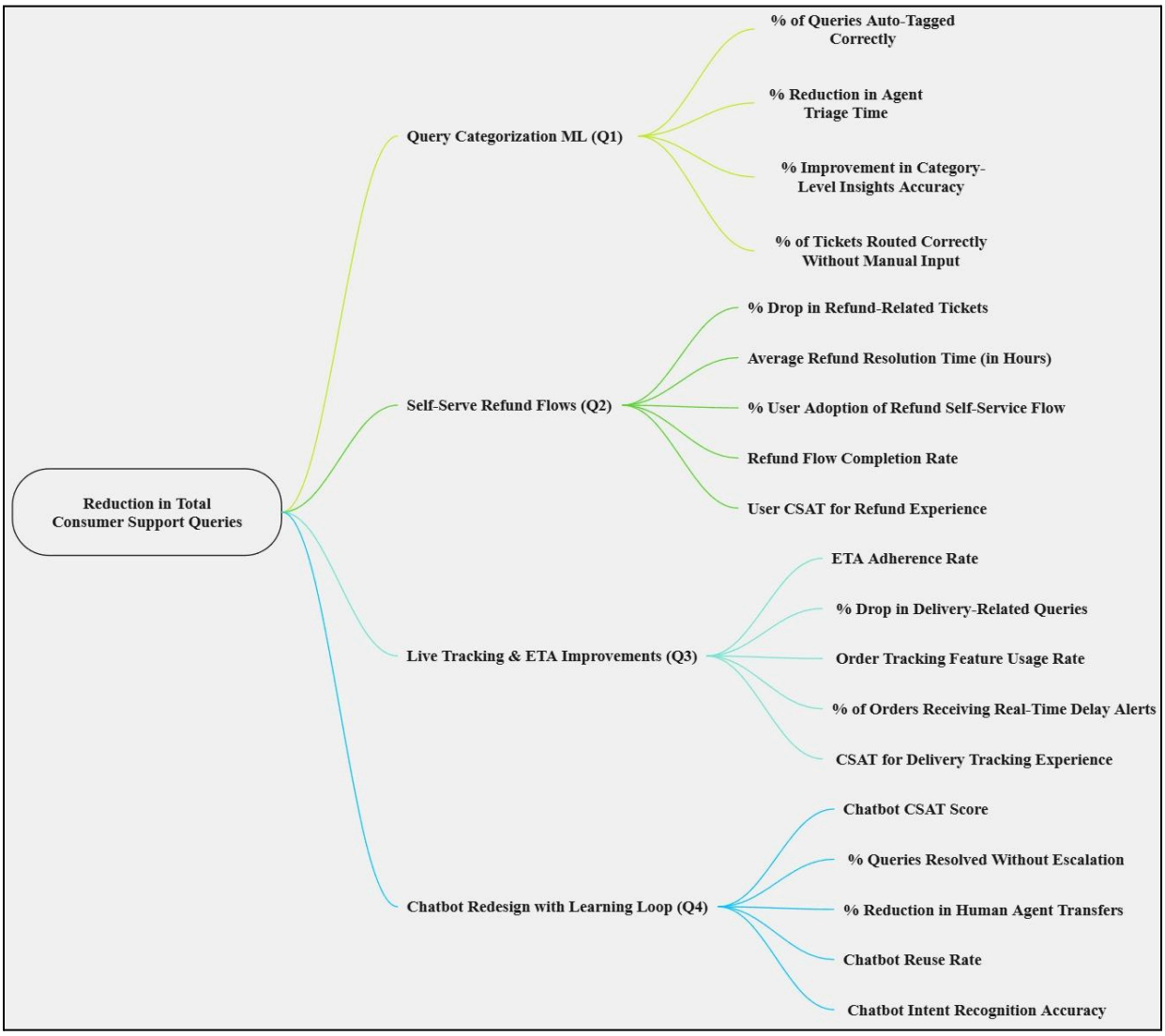
Each product outcome here is designed not just to resolve issues, but to preempt them, ensuring customer journeys are frictionless and predictable. Continuous iteration and data tracking will allow for prioritization and resource allocation based on the highest impact areas.

- **Proposed Product Roadmap :**

To ensure sustainable reduction in support query volume, Swiggy must execute product interventions in a phased manner. Each quarter will focus on solving one of the key root causes of support load using a data-backed, customer-first approach.

These initiatives have been prioritized based on query volume impact, ease of deployment, and potential for automation. The roadmap below lays out a quarter-wise strategy aligned to business objectives and operational readiness.

KPI Tree Based on Product Roadmap :



Quarterly Plan:

Quarter	Initiative	Product Outcome	Metric
Q1	Query categorization ML	Understand top issues	% of automated tagging accuracy
Q2	Self-serve refund flows	Reduce refund queries	% drop in refund tickets
Q3	ETA improvements & live tracking UI	Reduce delivery queries	ETA adherence rate
Q4	Chatbot redesign with learning loop	Improve automation	Chatbot CSAT, Resolution %

- ## Measurement Strategy :

Measurement strategy is vital for Swiggy to evaluate the effectiveness of its product-led support initiatives. By leveraging a blend of quantitative performance data and qualitative user insights, Swiggy can track improvements in customer satisfaction, monitor platform reliability, and optimize operational efficiency across its fast-scaling ecosystem.

1. Support Query Metrics:

Total Query Volume: Track overall and category-specific support queries to assess reduction trends.

Query Distribution by Type: Helps understand the shift in nature of support requests over time.

Query Deflection Rate: Measure the percentage of users who resolve their issues through self-service or automated flows without reaching agents.

2. Resolution Effectiveness:

First Contact Resolution (FCR) Rate: Percentage of queries resolved in the first interaction (via chatbot or agent).

Time to Resolution (TTR): Average time taken to close a support query (by type and channel).

Escalation Rate: Proportion of queries passed from chatbot to live agent, reflecting automation effectiveness.

3. User Experience Metrics:

User experience metrics are crucial to evaluating the impact of support-related product interventions on the end-user. These KPIs provide insight into how users perceive support features and how their satisfaction, trust, and loyalty are affected.

Metric	Definition	Q2 Benchmark Value	Target Trend
Customer Satisfaction (CSAT)	Avg. score collected post-resolution (1–5 scale)	**4.2**	**↑ 4.5**
Net Promoter Score (NPS)	% Promoters - % Detractors post-support experience	**38**	**↑ 45**
User Effort Score (UES)	Score on how easy it was to resolve issue (1–7 scale)	**5.8**	**↑ 6.2**
Resolution Path Success Rate	% of users who resolve issue within 2 steps	**67%**	**↑ 75%**
Repeat Contact Rate	% of users who contact again within 24–48 hrs	**12.50%**	**↓ 8%**
Time to Support Initiation	Avg. time from issue experience to support entry	**1 min 45 sec**	**↓ 1 min**
Self-Service Adoption Rate	% of users who resolve issues using self-serve flows	**43%**	**↑ 55%**
Chatbot Initiated Queries	% of total queries handled first via chatbot	**52%**	**↑ 60%**

4. <u>Operational Efficiency:</u>

Support Cost per Order: Calculated by dividing total support operating costs by total orders served.

Agent Utilization Rate: Evaluates how efficiently agent bandwidth is being used.

Support Headcount vs. Query Volume: Measures operational scalability with respect to support load.

5. <u>Product-Specific KPIs:</u>

Each initiative should have custom dashboards tracking:

- Query tagging accuracy (Q1)
- Refund flow completion rate (Q2)
- Delivery ETA adherence (Q3)
- Chatbot resolution rate and reuse (Q4)

6. <u>Continuous Monitoring & Feedback Loops:</u>

- A/B testing of support flows and in-app features to identify best-performing designs.
- Voice of Customer analytics to extract pain points from feedback.
- Weekly dashboards for internal review across product, operations, and CX teams.

- **Final Recommendation :**

Swiggy must adopt a product-led support strategy by embedding intelligence, automation, and continuous learning into its customer service ecosystem.

Each recommendation below is aimed at reducing dependency on human agents, improving the user experience, and building a scalable support infrastructure:

- **Invest in ML/NLP Systems to Understand Query Intent**:

  Swiggy should implement machine learning models that can classify, cluster, and extract intent from incoming support queries in real-time. This enables smarter routing to self-help, chatbot, or agents, reducing triage delays. Using historical data, the system can surface new problem patterns and support needs proactively.

- **Prioritize Self-Service Design Across App Touchpoints**:

  Introduce user-centric, guided support flows that allow customers to solve their issues independently. Features like smart FAQs, dynamic refund/cancellation flows, and in-context support widgets must be made easily accessible across order details, notifications, and chat interfaces. This reduces friction and builds user trust.

- **Embed Feedback Mechanisms Post-Resolution:**

  Collect structured feedback immediately after each support interaction to measure satisfaction, effort, and confidence in the resolution. Feedback loops—CSAT surveys, thumbs-up/down on chatbot answers, or open text forms—help detect usability issues and training gaps in automation systems.

- **Build a Support-Product Loop**:

  Enable bi-directional intelligence sharing between the support and product teams. Product teams should receive regular insights into user pain signals, top recurring issues, and unresolved gaps from support tickets. In turn, the support experience should evolve dynamically based on product changes, policy updates, and user journey improvements.

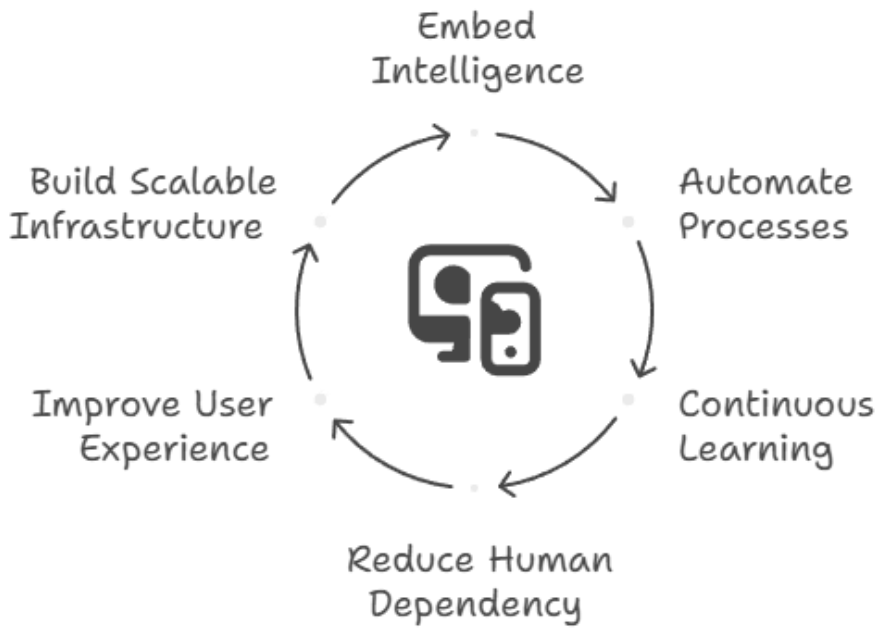


Figure : Product-Led Support Strategy

- **<u>Conclusions</u> :**

By aligning product development with key business outcomes, Swiggy can significantly reduce customer support load while enhancing user satisfaction and operational efficiency. A data-driven, product-first approach—focused on automation, self-service, and feedback—enables scalable support without compromising on quality. The roadmap and metrics outlined serve as a foundation for transforming customer service into a strategic differentiator for Swiggy.

The recommendations outlined—ranging from AI-powered query understanding to embedded self-service and feedback loops—represent a shift toward a proactive support model that learns continuously and adapts to user needs. With a structured roadmap and measurable KPIs, Swiggy can ensure cross-functional alignment across product, engineering, and support teams.

Ultimately, this transformation positions customer support not just as a reactive problem-solving unit, but as a strategic function that contributes to long-term retention, loyalty, and brand differentiation in a competitive market.

THANK YOU