

PM Interview Prep: 204 Questions with Answer Points

Product design, metrics, strategy, estimation, root-cause, technical and behavioral questions - each with a one-line framing and tight answer points built on standard PM frameworks.

Questions are grouped by type so each section doubles as framework practice. Numbering is continuous (1-204).

Framework cheat sheet (use these names in the interview)

CIRCLES	Design questions: Comprehend (goal/constraints) - Identify user - Report needs - Cut to priorities - List solutions - Evaluate trade-offs - Summarize.
AARRR (pirate metrics)	Acquisition - Activation - Retention - Referral - Revenue. Pick metrics from the stage the question cares about.
HEART	Happiness - Engagement - Adoption - Retention - Task success. Google's UX metric set.
North Star + guardrails	One metric that captures core value, plus counter-metrics that must not get worse (e.g. grow watch time without hurting retention).
Root-cause funnel	Diagnosis questions: 1) Verify the data/definition, 2) Segment the drop (platform, region, user type, time), 3) Internal causes (release, bug, price, policy), 4) External causes (competitor, season, news, regulation), 5) Confirm with data and fix.
Fermi estimation	Estimation questions: state assumptions, break into a multiplication chain, compute round numbers, sanity-check the result. The structure matters more than the number.
RICE / value vs effort	Prioritization: Reach x Impact x Confidence / Effort, or a simple value-versus-effort 2x2.
5 Whys / fishbone	Keep asking why until you reach a fixable cause; group causes into people, process, product, external.

Section 1 - Metrics & Measurement (Q1-Q39) - frameworks: AARRR, HEART, North Star + guardrails

Pattern: state the product's core value, pick one North Star, support with funnel metrics (AARRR or HEART), and always name a guardrail metric so growth in one number does not hide damage elsewhere.

1. What are the top 3 metrics you will track as a product head for Swiggy? **AARRR**

Food delivery is a three-sided marketplace (customer, restaurant, rider), so pick one metric per side of value.

- **Orders per day (completed deliveries)** - the North Star; captures total value created on the platform.
- **Order-to-delivery time (median, P90)** - the core experience promise; drives repeat orders.
- **Monthly order frequency per active user (retention)** - shows habit formation, not just one-time discounts.
- Guardrails: delivery partner earnings per hour (supply health), restaurant cancellation rate, contribution margin per order (unit economics).

2. What are the key metrics to be captured for a streaming service product? **HEART**

A streaming service wins when people find something to watch fast and keep coming back.

- **Engagement:** watch hours per user per week, % of sessions that result in 10+ minutes of playback.
- **Retention:** DAU/MAU ratio, month-1 and month-3 subscriber retention, churn rate.
- **Content success:** completion rate per title, % of viewing from recommendations vs search.
- **Task success:** time from app open to first play (aim under 30-60 seconds), search success rate.
- **Business:** CAC vs LTV, trial-to-paid conversion. Guardrail: buffering ratio / playback errors (QoE).

3. What data points would you track to know if a Zoom meeting was successful? **HEART**

Success = the meeting achieved its purpose with no friction; measure both the host and the attendees.

- **Join friction:** % of invitees who joined, average join time, % who needed a retry.
- **Quality:** audio/video drop rate, freeze count, average latency and jitter per participant.
- **Engagement:** % speaking time distribution, chat/reaction count, screen-share time, attentiveness proxy (camera-on %, tab-focus).
- **Outcome:** meeting ended on time, post-meeting survey (1-5 "was this worth it"), follow-up actions booked.
- **Task success:** 7-day re-meeting rate for recurring meetings; if people reschedule with Zoom again, it worked.

4. What metrics would you track to increase content consumption of Spotify by X times?

North Star + funnel

Consumption = listening time; grow it by removing friction between opening the app and pressing play.

- **North Star:** total listening hours per user per week.
- **Funnel:** sessions per week - % sessions with a play within 30s - tracks completed per session - playlist/algorithm share of plays.
- **Discovery:** % of plays from personalized surfaces (Discover Weekly, Daily Mix, autoplay radio) - the lever that scales consumption.
- **Experiments:** better autoplay continuation, offline pre-downloads, lock-screen resume; measure uplift per surface.
- Guardrails: skip rate (rising skips = worse recommendations), free-to-paid conversion, churn.

5. Define the metrics for YouTube search. Task success

Search succeeds when a query turns into satisfied watching quickly.

- **Relevance:** click-through rate on top-3 results, % of clicks with long watch time ("quality CTR", not just clicks).
- **Effort:** time to first click, % of queries needing a reformulation (a rewrite = search failed the first time).
- **Abandonment:** % of searches with zero clicks, exit rate from results page.
- **Outcome:** watch time per search session, % of searches leading to a subscribe/like/save.
- Guardrail: clickbait satisfaction - post-watch survey or like/dislike ratio on search-driven views.

6. You are the PM for Amazon's cart page. How would you build a dashboard to measure success? Funnel

The cart is the last step before money; the dashboard tracks how many carts turn into orders and what blocks them.

- **Core funnel:** cart views - checkout starts - payment success - order placed; conversion at each step, split by device.
- **Value:** average cart value, items per cart, attach rate of "save for later" moving back to cart.
- **Friction:** cart abandonment rate with reason codes (shipping cost shown, delivery date, payment failure), page load time.
- **Health:** price-change notifications acted on, out-of-stock-in-cart rate.
- Structure the dashboard: top-line KPI row, funnel viz, segment drill-downs (new vs returning, category, geo), alert thresholds.

7. Top six metrics for WhatsApp; pick one and give three improvements. AARRR + deep dive

Messaging is a habit product; depth of daily use matters more than downloads.

- **Six metrics:** DAU, messages sent per DAU, DAU/MAU stickiness, 30-day retention of new users, call minutes per user, group participation rate.
 - **Pick:** messages sent per DAU - it reflects core value and correlates with every other metric.
 - **Improvement 1:** richer replies (swipe-to-reply, inline polls) to make group conversation easier - measure replies per thread.
 - **Improvement 2:** smarter notification summaries so muted-group users re-enter conversations - measure mute-to-active rate.
 - **Improvement 3:** status/stories prompts to spark 1:1 replies - measure status-to-chat conversion.
- Guardrail: spam/report rate.

8. What metrics would you use to measure notifications at Facebook?

HEART + guardrails

Notifications are a re-engagement channel; measure both the pull back to the app and the annoyance cost.

- **Delivery funnel:** sent - delivered - seen - clicked; CTR by notification type.
- **Value:** % of clicks leading to a meaningful action (comment, reply), session length from notification entry.
- **Annoyance cost:** opt-out / mute rate, "turn off all" rate, dismiss-without-read rate - the key guardrails.
- **Efficiency:** incremental DAU lift per 100 notifications sent (does sending more actually add usage?).
- Segment by user type: heavy users need fewer, lapsed users need better timing.

9. How would you measure the performance and health of a Netflix Original?

Value chain

An Original is an investment; measure whether it acquires, engages and retains subscribers relative to its cost.

- **Reach:** % of subscribers who watched 2+ minutes in the first 28 days; first-week viewership rank.
- **Quality:** completion rate, binge velocity (episodes per day), rewatch rate, IMDb-style post-watch rating.
- **Business impact:** % of new signups who watch it in week 1 (acquisition driver), churn difference between viewers and matched non-viewers (retention driver).
- **Efficiency:** cost per completed view hour vs licensed content.
- Guardrail: does it cannibalize viewing of other high-value content or expand total watch time?

10. What is the most important metric to track in Google Search and why?

North Star

Search exists to answer questions; everything ladders up to successful answers.

- **Most important:** successful-query rate - % of queries where the user finds their answer (proxied by long-click on a result or no reformulation).
- Why not revenue: ads follow satisfied users; a search engine that answers badly loses the habit that monetizes.
- Support metrics: time to first click, reformulation rate, zero-result rate, queries per user per day.
- Guardrail: ad CTR vs organic satisfaction balance - don't let monetization degrade answer quality.

11. If you were the PM for Lyft, what dashboard would you build to track the health of the app?

Marketplace health

A ride marketplace is healthy when riders get cars fast and drivers earn well - the dashboard must show both sides.

- **Rider side:** open-to-request conversion, ETA at request, pickup wait time, ride completion rate, rating after ride.
- **Driver side:** online hours, utilization (% of hour with a passenger), earnings per hour, acceptance rate.
- **Marketplace balance:** supply/demand ratio by zone and hour, surge frequency and magnitude, cancellation rates (both sides).
- **Business:** rides per week, take rate, rider and driver 30-day retention.
- Layout: live ops view (map + ETA heatmap) on top, daily trends below, alerts on ETA or cancellation spikes.

12. You are on the Reddit PM team. How would you measure the success of Reddit?

Community value

Reddit's value is community-generated content and discussion, so measure contribution, not just consumption.

- **Consumption:** DAU/MAU, time spent, posts viewed per session.
- **Contribution:** % of users who post or comment weekly, upvotes per user - the ratio of creators to lurkers is Reddit's engine.
- **Community health:** subscriber growth of subreddits, mod actions per report, % of comments removed for toxicity.
- **Retention:** week-4 retention of new users, % of users subscribed to 3+ communities (a known retention driver).
- **Business:** ad revenue per DAU without hurting the above guardrails.

13. How would you measure the success of the Netflix recommendation engine?

Task success

The engine succeeds when people watch what it suggests and stay because of it.

- **Direct:** % of total watch time coming from recommendations (vs search/browse), CTR on recommended rows.
- **Quality:** completion rate of recommended titles vs non-recommended, post-watch thumbs-up rate.
- **Efficiency:** time to first play from app open - good recommendations shorten the "what do I watch" hunt.
- **Long-term:** retention/churn difference between users who engage with recommendations and those who don't (A/B holdout).
- Guardrail: catalog diversity - % of watch time on top-100 titles; recommendations shouldn't collapse into a bubble.

14. LinkedIn: which top three metrics would you check first thing in the morning?

Daily ops

A morning check looks for breakage and trend shifts, not deep analysis.

- **DAU (vs 4-week same-weekday average):** catches outages and trend breaks immediately.
- **Feed engagement rate:** % of DAU liking/commenting/posting - the core value loop.
- **Key revenue pulse:** job applications submitted or recruiter InMails sent - the money side of the marketplace.
- Then drill: any metric off by more than 2 standard deviations gets a segment breakdown (platform, geo, release version).

15. You are a PM for Facebook Events focused on growth. What would be your North Star metric? North Star

Events create value only when they actually bring people together.

- **North Star:** weekly attendees - number of users who attend (check in / confirm attending) an event each week.
- **Why:** RSVPs can be gamed and page views are hollow; actual attendance captures real value delivered to both organizers and attendees.
- **Input metrics:** events created per organizer, invite-to-RSVP rate, RSVP-to-attendance rate, discovery share (events found via feed/suggestions).
- **Guardrail:** event quality - post-event rating or % of events cancelled, so growth doesn't come from spammy events.

16. Pick a feature and tell us how you would assess its success at launch. Launch template

Example: Instagram's "Close Friends" list - a privacy feature for sharing Stories with a smaller circle.

- **Before launch:** define the hypothesis (people share more when audience control is easier) and set baseline metrics.
- **Adoption:** % of active users who create a Close Friends list within 30 days.
- **Behavior change:** stories posted per user per week among adopters vs matched non-adopters (the real test).
- **Quality:** list edit frequency, % of stories restricted to Close Friends, viewer reply rate on those stories.
- **Guardrails:** overall story views shouldn't drop (smaller audience), unfollow/block rate stable. Decide scale/iterate/kill after 4-6 weeks.

17. How would you design a dashboard of call quality for WhatsApp? QoE dashboard

Call quality is invisible until it breaks; the dashboard must catch problems by network, device and region.

- **Quality KPIs:** call drop rate, setup failure rate, average MOS-style audio score, freeze/latency/jitter P90.
- **Experience KPIs:** call completion rate, average call duration, % of calls rated (post-call rating), switch-to-video success.
- **Dimensions:** slice by country, carrier, network type (WiFi/4G/2G), device class, app version - quality bugs hide in segments.
- **Structure:** real-time alerting view (drop-rate spikes) + daily trend view + release-comparison view (did the new version regress quality?).
- **Tie to outcome:** % of users who made a call last week and returned to call again.

18. You are the PM of the "Share" feature on Facebook. Measure its success. Feature funnel

Share succeeds when content spreads and both the sharer and the receiver get value.

- **Usage:** shares per DAU, share-button CTR, % of posts shared at least once.
- **Propagation:** downstream engagement on shared posts (clicks, comments from recipients), re-share depth (viral chain length).
- **Intent mix:** share-to-feed vs share-to-Messenger vs copy-link - private sharing growth matters even if less visible.
- **Health:** originality guardrail - % of shares that are original content vs low-quality viral bait; report/hide rate on shared posts.
- North Star for the feature: weekly successful shares (a share that got at least one meaningful recipient interaction).

19. What is the most important metric for Slack and why? North Star

Slack is a team habit tool; value exists only when teams communicate there daily.

- **Most important:** weekly active teams (teams with sustained message activity), not individual users - Slack is bought and churned as a team.
- Supporting: messages sent per active user, DAU/MAU stickiness, % of messages in channels vs DMs (public knowledge-sharing = more lock-in).
- Activation: % of new teams reaching ~2,000 messages (Slack's known stickiness threshold).
- Business linkage: active teams - paid conversion - expansion seats; the North Star predicts revenue.

20. Users have decision fatigue on Uber Eats. What success metrics would you use to check the impact (of fixes)? Task success

Decision fatigue shows up as long browsing with no order; fixes should shorten the path to a confident choice.

- **Primary:** browse-to-order conversion rate, and time from app open to cart add.
- **Friction signals:** restaurant pages viewed per order, filter usage, back-navigation count, session abandonment after 3+ minutes of browsing.
- **Choice quality:** reorder rate (repeat orders = confident past choice), post-order rating, refund/complaint rate.
- **Experiment readout:** if you add "top picks for you" or smaller curated menus, measure A/B uplift in conversion and time-to-cart.
- Guardrail: average order value and restaurant diversity of orders - curation shouldn't shrink basket or concentrate all orders in 3 restaurants.

21. How would you measure the success of Google Calendar? **HEART**

Calendar succeeds when it is the trusted source of your time - people plan in it and miss nothing.

- **Adoption/Engagement:** weekly active users, events created per user per week, % of events with 2+ attendees (collaborative use).
- **Task success:** time to create an event, % of meeting invites accepted/declined within 24h, "find a time" feature usage.
- **Reliability value:** notification open-to-event-start rate, missed-meeting proxy (no-shows on video links).
- **Happiness:** CSAT/NPS surveys, feature-level satisfaction (scheduling assistant, working location).
- Guardrail: churn to other calendars is slow to see, so watch workspace attach rate and cross-app usage (Gmail/Meet integration).

22. How do you define success for Yelp reviews? **Two-sided value**

Reviews succeed when readers trust them enough to choose a business and writers feel heard.

- **Reader value:** % of business-page visits that read reviews, "useful/funny/cool" vote rate, conversion to direction request or call after reading.
- **Writer value:** % of reviews getting a vote within 7 days, repeat-reviewer rate (monthly active reviewers).
- **Quality/Trust:** % of reviews filtered as fake/suspicious, review length and photo attach rate, distribution balance (not all 1 or 5 stars).
- **Coverage:** % of listed businesses with 5+ recent reviews - freshness matters.
- Guardrail: review fraud detection precision; too aggressive filtering kills genuine reviewers' motivation.

23. How would you measure the success of Uber Ride? **Marketplace**

A ride is successful when a request becomes a safe, fast, completed trip - for rider and driver.

- **North Star:** completed trips per week.
- **Rider experience:** open-to-book conversion, average pickup ETA, cancellation rate, post-ride rating.
- **Driver experience:** earnings per online hour, utilization, acceptance rate, driver 30-day retention.
- **Efficiency:** % of requests fulfilled within 5 minutes, surge incidence, cost per trip.
- Guardrails: safety incidents per 10k trips, rider churn after a bad trip (ETA >10 min or driver cancel).

24. How would you measure the success of Gmail? **HEART**

Email succeeds when people reach inbox zero fast and never miss what matters.

- **Engagement:** DAU/MAU, sessions per day, emails read/sent per active user.
- **Task success:** time to triage inbox (open to zero-unread or batch-archive), search success rate, spam false-positive rate (good mail in spam is the worst failure).
- **Adoption:** usage of value features - labels/filters, snooze, smart reply acceptance rate.
- **Happiness:** NPS, % of users on default settings vs custom (customization = investment).
- Guardrail: spam catch rate vs false positives, storage growth per user (cost side).

25. How would you evaluate the success of Reactions on Facebook? Feature launch

Reactions (Love, Haha, Wow, Sad, Angry) added expressive range beyond Like; success = richer expression without replacing it.

- **Adoption:** % of post-interactions using reactions vs plain Like, per-user reaction usage per week.
- **Expression value:** % of posts receiving 2+ distinct reaction types (shows emotional range is used), reaction-to-comment conversion (does a Sad prompt supportive comments?).
- **Ecosystem:** change in total post feedback (reactions should add interactions, not cannibalize comments).
- **Ranking value:** do reaction types improve feed ranking quality (long-term session satisfaction in A/B holdout)?
- Guardrail: negative-reaction misuse (Angry brigading), harassment reports on reacted posts.

26. You are the PM of Facebook Timeline. How would you measure retention? Retention

Timeline retention = do people keep returning to profiles, and do profile owners keep posting?

- **Viewer side:** % of DAU visiting any timeline, repeat profile-visit rate within 7 days, time spent on profiles.
- **Owner side:** % of users posting to their own timeline per week/month, profile-update frequency.
- **Cohort retention:** classic N-day retention curves (D1/D7/D30) of timeline visitors, split by new vs existing users.
- **Depth:** scroll depth on a profile, % of visits reaching older posts (shows archival value).
- Guardrail: privacy controls usage - if owners lock down timelines, viewer metrics fall; track both together.

27. How would you evaluate the success of Instagram Stories? Creator-consumer loop

Stories work when creation is effortless and viewing is habitual - measure both sides of the loop.

- **Creation:** % of DAU posting a story, stories per creator per day, creation-tool usage (stickers, filters) as a proxy for effortlessness.
- **Consumption:** % of DAU viewing stories, stories viewed per session, completion rate through the story tray.
- **Interaction:** replies/reactions per story, DM conversations started from stories (the social glue).
- **Business:** ad impressions in stories, ad CTR, incremental time spent vs feed.
- Guardrail: does stories growth cannibalize feed engagement in a way that hurts overall retention?

28. How would you measure the success of Google Photos? **HEART**

Photos succeeds when people trust it with their memories and find any memory instantly.

- **Trust/Adoption:** % of users with backup turned on, photos backed up per user, cross-device sync success rate.
- **Engagement:** weekly active users, "Memories" view and share rate, search usage per week.
- **Task success:** search-to-photo-found rate (people, places, things queries), time to find a specific photo.
- **Retention:** 30/90-day retention; photos stored is a switching cost - watch upload continuity as the loyalty signal.
- Guardrail: storage quota hit rate and upgrade-to-paid conversion without degrading free trust.

29. What are the success metrics for Facebook Live at the growth phase? **Growth stage**

At growth stage, prioritize adoption and supply of broadcasts before monetization.

- **Supply:** broadcasts started per day, % of eligible creators/pages going live weekly, broadcast length distribution.
- **Demand:** live viewers per broadcast, peak concurrent viewers, % of viewers watching 1+ minute.
- **Engagement:** comments/reactions per viewer-minute (live's edge over recorded video), share rate.
- **Habit:** repeat broadcaster rate week-over-week, viewer return rate for the same broadcaster.
- Guardrails: stream quality (start-up time, rebuffering), content-policy violation rate - growth can't outrun moderation.

30. What metrics track improvement of sense of community within Facebook Groups?

Community health

Community is felt, so proxy it through reciprocity and belonging signals, not just activity volume.

- **Reciprocity:** % of posts getting a comment within 1 hour, comment-thread depth (back-and-forth = conversation, not broadcast).
- **Belonging:** % of members active weekly, new-member first-post rate within 7 days, % of members with 3+ interactions with the same person (relationships forming).
- **Recognition:** reactions given per member, "top contributor" badge engagement, member-to-member tag frequency.
- **Safety/belonging:** report rate, % of members who mute or leave (inverse signal), admin action volume.
- Survey layer: periodic "do you feel this group is a community?" poll as ground truth to calibrate the proxies.

31. How would you improve Facebook Groups and measure whether you were successful?

Improve + measure

Pick the biggest gap first: most members are lurkers; improvements should convert reading into participating.

- **Improvements:** new-member onboarding prompts (introduce-yourself templates), daily digest of top posts to re-pull members, admin tools for scheduled posts and member Q&A, better discovery of related groups.
- **Success metrics:** active-member rate (% of members acting weekly), first-post-within-7-days rate for joiners, comment depth per post.
- **Retention:** 30-day group retention of new members, group-level churn.
- **Method:** A/B each improvement on a % of groups, read at 2 and 4 weeks, keep only what lifts participation without raising report rate.

32. Marketplace PM: sellers don't mark items as sold. What functionality and metrics determine whether an item has likely sold? **Inference design**

If sellers won't tell you, infer the sale from behavior signals - then measure the inference quality.

- **Signals:** message exchange between buyer and seller (price/meetup words), listing views stopping abruptly after messages, seller deleting or going inactive on the listing, both parties' location overlap.
- **Functionality:** one-tap "Mark as sold" prompt after detected buyer chat, auto-archive suggestions, "did you sell this?" follow-up with reward (better ranking for responsive sellers).
- **Metrics:** % of listings resolved (sold/withdrawn) within 30 days, precision of likely-sold classifier (validated by the follow-up prompt's answers), stale-listing rate.
- **Why it matters:** stale listings waste buyer time; measure buyer contact-to-purchase conversion before/after cleanup.

33. What success metrics would you use to measure a payment gateway product?

B2B metrics

A payment gateway is infrastructure: merchants buy reliability and conversion, not features.

- **Reliability:** transaction success rate (target 95%+ for domestic cards), uptime, average API latency, error rate by failure code.
- **Merchant value:** checkout conversion uplift vs previous gateway, supported payment-method coverage, settlement time (T+n days).
- **Adoption:** active merchants, transaction volume (TPV) and count per merchant, integration time from signup to first live transaction.
- **Trust:** fraud catch rate vs false-decline rate, chargeback rate, dispute resolution time.
- **Business:** take-rate revenue, merchant churn, net revenue retention.

34. What is the most important metric for Uber Eats and why? North Star

Food delivery's core promise is a completed, on-time meal - one metric captures most of it.

- **Most important:** completed on-time deliveries per day (or per week).
- Why: it blends demand (orders), supply (couriers available), marketplace balance (restaurant prep), and experience (speed) into one number; a rise means all three sides worked.
- Support: order frequency per customer, courier utilization, restaurant prep-time accuracy.
- Guardrails: cost per delivery (don't buy growth with unsustainable subsidy), courier earnings per hour.

35. What is the most important metric for Google Docs and why? North Star

Docs is a collaboration tool; solitary typing is table stakes, collaboration is the moat.

- **Most important:** weekly collaborative documents - documents edited by 2+ people (or shared) per week.
- Why: collaboration is why teams standardize on Docs; it drives workspace lock-in and differentiates from local editors.
- Support: DAU, time in docs, comment/suggestion volume per doc, real-time co-editing session count.
- Guardrail: single-user creation must stay healthy too - collaboration starts with someone creating.

36. Identify one metric for Facebook Marketplace that the whole team should monitor daily.

North Star

Marketplace value happens when a listing turns into a real exchange.

- **One metric:** daily completed-transaction proxies - buyer-seller conversations that reach an exchange signal (e.g. "sold" marking or meetup confirmation).
- Why this one: listings (supply) and views (demand) are inputs; a team watching the value event daily catches marketplace balance problems fastest.
- Watch alongside (not as the daily metric): new listings per day, time-to-first-buyer-message, % of listings getting zero messages in 7 days.
- Segment the daily number by category and city - marketplaces fail locally before they fail globally.

37. What metrics would you look at as PM for Instagram ads? Two-sided ads

Ad products must serve advertisers (returns) and users (experience) at once - measure both.

- **Advertiser side:** CTR, conversion rate (install/purchase/lead), cost per action, return on ad spend, advertiser retention and budget growth.
- **User side (guardrails):** ad load (% of feed/stories that are ads), ad hide/report rate, session-length impact of ad exposure in holdout tests.
- **Platform:** revenue per DAU, fill rate, auction density (bidders per impression - low density = weak demand).
- Balance rule: grow revenue per DAU only while ad-hide rate and session time stay flat; otherwise pull ad load back.

38. How would you measure the success of Facebook Likes? Feature health

Like is the lightest feedback loop; it works if it keeps creators posting and viewers acknowledging.

- **Volume:** likes per DAU per day, % of viewed posts receiving a like.
- **Loop value:** % of posters who receive a like within 1 hour (fast feedback drives the next post), correlation of received likes with 7-day repost rate.
- **Distribution:** Gini-style spread - likes concentrated on 1% of posts signal a rich-get-richer problem for casual users.
- **Ranking input:** how much a like predicts long-term session satisfaction vs a comment (recalibrate like weight in feed ranking).
- Guardrail: like-seeking bait prevalence; hidden-like experiments measure whether counts distort posting behavior.

39. What metrics measure the success of the Save feature at Facebook? Feature funnel

Save is an intent signal: "this is worth coming back to" - success is both saving and actually returning.

- **Usage:** saves per DAU per week, % of viewed content saved, save-button discoverability (CTR on the save entry point).
- **Return value (the real test):** % of saved items revisited within 7 days, saved-collection session frequency.
- **Outcome:** for saved Marketplace items/events/links, conversion to purchase/attendance/read vs non-saved baseline.
- **Ranking value:** save as a quality signal in feed ranking vs like (saves indicate deeper intent).
- Guardrail: save-and-never-return rate - if 90% of saves are never revisited, the feature is a junk drawer and needs resurfacing (reminders, digest).

Section 2 - Diagnosis & Root Cause (Q40-Q65) - framework: root-cause funnel (verify data - segment - internal - external - confirm)

Pattern for every "metric dropped/spiked" question: never jump to a fix. First check the number is real, then find WHERE it moved, then test internal and external causes, then act.

40. Facebook's DAU is decreasing 15% per week for 3 weeks. Find out why. Root-cause funnel

A sustained 15%/week decline is severe - treat it like an incident, not a trend.

- **Verify:** is logging intact? A telemetry or definition change can fake a drop. Compare with independent signals (ad impressions, server requests).
- **Segment:** which platform (Android/iOS/web), region, age group, new vs returning users is falling? A drop concentrated in one segment points at the cause.
- **Internal:** releases in the last 3 weeks (performance, login bugs, feed ranking change), notification system outage, policy/moderation changes.
- **External:** a country-level block or app-store removal, a competitor launch, negative press cycle, seasonal pattern (compare year-over-year, not week-over-week).
- **Confirm & act:** match the cause to the segment evidence, roll back or fix, and set a recovery target with weekly review.

41. Spotify installs increased 25%. Give at least four reasons. Growth diagnosis

A spike is also a diagnosis problem - find which lever moved before celebrating.

- **Marketing:** a new ad campaign, influencer push, or app-store featuring - check install curve against campaign dates.
- **Product:** a launch (new free tier, podcast content, Wrapped season), a viral sharing feature.
- **External:** a competitor outage or controversy, a bundled partnership (phone maker, telco pre-install), seasonal effect (New Year, festival season).
- **Data artifact:** measurement change, bot/fraud installs, re-attribution of existing users.
- Next step: segment installs by source, country and quality (do the new installs activate and retain?) - a spike of low-quality installs is not a win.

42. 50% of Paytm transactions were reduced last week. What could be the reasons?

Root-cause funnel

A 50% payments drop is almost always a failure somewhere, not vanished demand - start with reliability.

- **Technical:** gateway/bank partner downtime, a failed release, UPI rail issues, app crash spike - check success rate by payment method and bank.
- **Regulatory:** an RBI/banking restriction, KYC enforcement blocking users - payments is heavily regulated; rule this out early.
- **Product/policy:** fee introduction, removed cashback, a checkout flow change adding friction.
- **External:** competitor promotion pulling volume, a merchant category outage (e.g. a big biller integration broke), seasonality (post-festival trough).
- Verify instrumentation first; then segment by method, merchant category, region, and user tenure.

43. A large number of drivers are dropping out of a particular city. Why? Supply-side RCA

Driver churn is a supply problem - follow the earnings.

- **Earnings:** did pay per hour fall (rate-card change, bonus removal, fewer rides per hour from oversupply)? Compare driver earnings trend vs other cities.
- **Demand:** did ride volume drop in that city (season, event ended, competitor pricing war)?
- **Costs:** fuel price spike, vehicle financing change, new city regulation/permit cost.
- **Experience:** unsafe zones, rider-behavior issues, app bugs in that locale, better offer from a competitor onboarding drivers.
- Data to pull: driver-level earnings before exit, exit survey if any, hours-online trend. Fix: targeted incentives are a bandage; fix the structural earnings gap.

44. There's a 50% drop in Uber rides per day. Diagnose the problem. Root-cause funnel

Same funnel, marketplace version: a ride needs a rider, a driver, and a working app - find which broke.

- **Verify:** logging change vs real drop; compare app opens to ride requests (opens fine + requests down = pricing/supply issue; opens down = app/acquisition issue).
- **Segment:** city, time of day, product tier (Go/Premier), new vs existing riders.
- **Internal:** price increase, surge algorithm bug, driver exodus (check ETAs - long ETAs kill conversion), app crash in the latest release.
- **External:** competitor discount blitz, public-transit change, weather/event, regulation or strike, negative press.
- Confirm by matching segment pattern to cause; example: drop only at airport + flat ETAs = a policy change at the airport, not a pricing problem.

45. Creation of Facebook user groups is down 20%. What will you do? Root-cause funnel

Group creation is a supply-side metric - a drop usually means friction or weakened motivation for creators.

- **Verify & segment:** which surface (mobile/web), region, creator type (new vs established) declined?
- **Internal:** creation-flow change (added steps, naming requirements), a discoverability change making new groups fail to get members (so creators stop trying), moderation crackdown discouraging creation.
- **External:** saturation (most niches already have a group), shift to other platforms (Discord, WhatsApp communities).
- **Diagnose before acting:** if new groups fail to attract members, fix distribution (suggest new groups in feed) rather than the creation button.
- Track: creation rate recovery AND 30-day survival of new groups - creating more dead groups is not success.

46. Uber Eats orders are down. How will you track the root cause? Root-cause funnel

Decompose orders: $\text{orders} = \text{users} \times \text{sessions} \times \text{conversion}$ - the drop lives in one of them.

- **Funnel decomposition:** is traffic (opens) down, or conversion (open-to-order) down? Traffic down = marketing/season/competitor. Conversion down = pricing, fees, ETA, or availability.
- **Supply side:** restaurant count online, courier availability affecting ETAs - long ETAs suppress orders.
- **Segment:** city, cuisine, time of day, customer tenure; a lunch-only drop is a different problem than an everywhere drop.
- **Internal changes:** delivery-fee or small-cart-fee change, app release, promo budget cut (orders propped by coupons fall when coupons stop).
- **External:** competitor promotion, weather, festival/fasting periods, macro spending pullback. Confirm with YoY comparison and a holdout-city comparison.

47. Walmart's order return rate is increasing. What would you look into? RCA by category

Returns are a margin killer; isolate whether the product, the promise, or the buyer behavior changed.

- **Segment:** which categories, sellers (first-party vs marketplace), regions, and return reasons grew? Returns concentrate - find where.
- **Product quality:** new supplier or marketplace sellers with misrepresented listings; check reviews and "not as described" reasons.
- **Expectation gap:** sizing/photo/description issues (apparel), damaged-in-transit (packaging/carrier change).
- **Policy effect:** a more generous return policy predictably raises returns - was one introduced?
- **Fraud/behavior:** wardrobing (buy-use-return), serial returners. Actions: fix listing accuracy, tighten packaging, flag abusive accounts, charge restocking where fair - while protecting the trust that easy returns create.

48. Google Maps had a 10% drop in DAU. What do you do? Root-cause funnel

Maps usage tracks movement of people, so check both product causes and real-world behavior shifts.

- **Verify:** logging, definition change, time-zone artifacts in the DAU computation.
- **Segment:** platform (Android vs iOS), country, feature surface (navigation vs explore vs search).
- **Internal:** release bug (crash on start, broken location permission prompt), battery-drain regression, data-usage concern, UX change hiding the core action.
- **External:** seasonal (holidays = less commuting), weather events, a competitor (Apple Maps improvement on iOS), fuel prices reducing driving.
- **Act:** if a segment + release correlate, roll back; if seasonal, benchmark against last year and monitor recovery.

49. Late deliveries lead to churn (hypothesis). What data proves it for a food delivery app?

Hypothesis testing

Don't assume - show that lateness causes churn, not just correlates with it.

- **Correlational:** compare 30-day repeat rate of customers whose last order was late (say >15 min over promise) vs on-time; control for cuisine, city, order value, customer tenure.
- **Dose-response:** does churn rise with lateness (5 min late vs 30 min late)? A gradient supports causation.
- **Confounders:** late orders may cluster in rain/peak hours which also affect experience - control for weather and day-part.
- **Strongest evidence:** a natural or real experiment - compensate a random half of late-order customers (credit + apology) and see if their churn falls to on-time levels; if yes, lateness caused churn and the fix pays for itself.
- Also check: churn after first-ever late delivery vs after repeated ones (threshold effects).

50. Facebook Messenger usage is down. Figure out why and what you'd do. Root-cause funnel

Usage = opens x messages per open; find which component fell and in which segment.

- **Verify & segment:** platform, region, age; is it opens (people not coming) or depth (coming but messaging less)?
- **Cannibalization check (key for Meta):** did messaging shift to WhatsApp, Instagram DMs, or in-Facebook chat? Total company messaging may be flat while Messenger falls - that changes the response entirely.
- **Internal:** app bloat/performance complaints, notification changes, a redesign backlash.
- **External:** competitive shift (iMessage among teens), regional trends.
- **Actions depend on cause:** if teens moved to IG DMs, integrate rather than fight; if performance, invest in speed; measure recovery in messages per DAU, not installs.

51. About 1 million inactive Netflix users. What would you do about them? Win-back

Inactive but paying users are a churn time-bomb and a win-back opportunity - segment before spending.

- **Segment the inactive:** paying-but-not-watching (cancel soon), paused accounts, signed-up-never-watched (activation failure), seasonal watchers.
- **Diagnose why:** exit surveys, support tickets, content-gap analysis (did their favorite show leave?), price-increase timing.
- **Win-back plays:** personalized "here's what you missed" emails/notifications with one strong title, pause-instead-of-cancel option, price-appropriate plan downgrade offer (cheaper than losing them).
- **Never-watchers:** fix onboarding - first-session content match within 3 clicks.
- Measure: reactivation rate per campaign, and 90-day retention of reactivated users (a reactivation that churns again in a month is rented, not earned).

52. Instagram DAU dropped 20%. Steps to identify the root cause? Root-cause funnel

Same funnel, social app: verify, localize, internal, external, confirm.

- **Step 1 - verify:** analytics pipeline intact? Cross-check with server-side session counts and push-delivery logs.
- **Step 2 - localize:** platform, OS version, country, age cohort, new vs returning, surface (feed vs stories vs reels usage).
- **Step 3 - internal:** releases in the window (crash rate, startup time, feed algorithm change, notification volume change), outage history.
- **Step 4 - external:** competitor moment (TikTok feature), app-store ranking/feature change, press cycle, seasonality (check YoY).
- **Step 5 - confirm & act:** match evidence to one hypothesis, fix or roll back, define recovery metric and review cadence; document for the postmortem.

53. Spotify's 1-month trial users are not converting to paid. What product changes would you make? **Conversion RCA + fixes**

Trial conversion fails when users never feel the paid value or the free tier is "good enough".

- **Diagnose first:** split trial users by engagement - low-engagement trials (never formed a habit) vs engaged non-converters (habit but no willingness to pay). Different problems, different fixes.
- **Low engagement:** improve trial onboarding - playlist creation in first session, daily mix from day 1, download prompt so they feel offline value.
- **Engaged non-converters:** trial-limit reminders surfaced at moments of pain (ad breaks, skip limits), a mid-trial "you've heard X ads this week" message, family/student plan offers, discount for converting before trial ends.
- **Structural:** check free-tier generosity - if free is too good, conversion needs feature differentiation (offline, quality, skips), not just nagging.
- Measure: trial-to-paid by cohort, and 3-month retention of converters (bought with discount must still stick).

54. There has been a 15% drop in usage of Facebook groups. How would you fix it? **RCA then fix**

Usage drop in groups splits into supply (admins posting) and demand (members visiting).

- **Diagnose:** is the drop in group visits (demand), posts (supply), or interactions? Segment by group size, topic, member tenure.
- **Likely causes:** feed algorithm deprioritizing group content, notification changes, admin burnout (fewer posts - quieter groups - fewer visits: a loop), member shift to other surfaces.
- **Fixes by cause:** if discovery - resurface active groups in feed and notifications; if supply - admin tools (scheduled posts, content prompts, admin assistant); if engagement - prompts to comment, polls, events inside groups.
- **Loop awareness:** groups are a cold-start ecosystem; fix the side that starts the flywheel (usually admin activity).
- Measure recovery in weekly active group members and posts per active group, not raw group count.

55. New FB signups are not returning after 30 days. Identify the problem and solve it. **Activation funnel**

New-user churn is an onboarding problem: they never reached the "aha" moment (friends + content).

- **Diagnose:** build the day-0 to day-30 funnel - signup - friend adds - first feed session - first interaction - return visits. Find the cliff.
- **Common causes:** empty friend graph (no content worth returning for), empty/low-quality first feed, notification permission not granted (nothing pulls them back), confusing first-run experience.
- **Fixes:** friend suggestions from contacts during signup (Facebook's classic "10 friends in 14 days" insight), interest-based follow suggestions so day-1 feed has content, progressive onboarding (don't demand everything upfront), well-timed re-engagement notifications about friend activity.
- **Experiment:** A/B each fix on new cohorts; success metric is D30 retention of the cohort, with friend-count and session depth as leading indicators.

56. FB Events: overnight 10% drop in RSVPs. What happened? Root-cause funnel

"Overnight" points to something that changed suddenly - a deploy, an outage, or a seasonal boundary.

- **Verify:** logging change? Check against independent signals (event page views, notification sends).
- **Localize:** which surface (feed vs notifications vs shares), region, event type, device?
- **Recent changes:** deploy in the last 48h affecting the RSVP button, notification delivery (RSVPs are heavily notification-driven - check delivery rates), ranking change reducing event distribution.
- **External:** holiday/weekend boundary, a major news event crowding attention, daylight-saving/timezone bugs (classic for events).
- Confirm by matching the drop's shape (cliff vs slope) to a deploy timestamp; rollback is often the fastest test.

57. Weekly active users (WAU) for an iPhone app dropped. What happened? Root-cause funnel

Platform-specific framing: on iOS, check OS updates and App Store mechanics alongside the usual causes.

- **Verify:** analytics SDK change, iOS privacy prompt (ATT-style) breaking measurement - WAU may be undercounted, not down.
- **Segment:** iOS version (did a new iOS release break something?), device model, acquisition cohort vs existing users.
- **Internal:** app update with crashes or removed features, notification permission changes, subscription price change.
- **External:** App Store featuring lost, search ranking drop, competitor launch, seasonality.
- Distinguish "users left" from "users invisible": compare server-side activity to client analytics before any product change.

58. Instagram PM: MAU constant but DAU declined. What will you do? Frequency problem

MAU flat + DAU down = people still come monthly but less often - a habit/engagement problem, not a growth problem.

- **Confirm the shape:** DAU/MAU ratio is falling; check session frequency per user and which visits disappeared (daily casual scrollers first).
- **Diagnose:** content freshness (feed feels stale), notification fatigue leading to opt-outs (nothing pulls them back), competing habits (Reels time shifting to TikTok), algorithm change showing less relevant content.
- **Actions:** improve daily-visit hooks (stories from close friends first, better notification relevance and timing), re-engagement surfaces (digest, "while you were away"), measure by cohort.
- **Guard against:** don't fix frequency with notification spam - watch opt-out rate as the guardrail.
- Success metric: DAU/MAU (stickiness) recovering to baseline within 2-3 release cycles.

59. Spike in user-uploaded videos in June. Cause, and how would you test it?

Anomaly diagnosis

A spike needs an explanation as much as a drop - it may be real growth, an event, or garbage.

- **Candidate causes:** seasonal (school holidays = more uploads), a viral challenge/trend, a new upload feature or reduced friction, a marketing campaign, spam/bot uploads, a logging change.
- **Test with segmentation:** is the spike concentrated in certain geos, age groups, topics, or new accounts? Real seasonal growth is broad; spam is concentrated and low-quality.
- **Quality check:** views-per-upload, % removed by moderation, average video length - if quality metrics crashed while volume spiked, suspect spam.
- **Confirm:** compare against last June (YoY), and run a difference-in-differences: did similar platforms/regions spike too?
- Act accordingly: ride a real trend (feature it), or tighten upload defenses if it's abuse.

60. More Uber drop-offs at the airport than pick-ups. Why, and what would you do?

Marketplace imbalance

An airport imbalance is about where drivers choose to be and what alternatives riders have.

- **Why more drop-offs:** people take Uber TO the airport but choose taxis/shuttles/family pickups FROM it (price sensitivity on return, airport Uber surcharges, long queue/wait, confusing pickup zones).
- **Driver side:** airport pickup may be unattractive - long queue wait unpaid, complex staging lots - so supply avoids it, raising ETAs and pushing riders to alternatives (a loop).
- **Data:** request-to-fulfillment rate at airport vs city, pickup ETA, queue length, rider cancellation rate at pickup, price comparison vs taxi.
- **Actions:** improve pickup-zone wayfinding in-app, scheduled airport pickups, queue-priority or guaranteed-fare incentives for drivers, PIN/zone-matched pickups to cut search time.
- Measure: pickup fulfillment rate and rider repeat usage for airport trips.

61. 15% drop in the open rate of the Instagram app. What could have happened?

Root-cause funnel

Open-rate drops are usually pull-mechanism failures (notifications) or friction at app start.

- **Verify & segment:** platform, OS version, region, user tenure; check push delivery and open rates from notifications specifically.
- **Notification causes:** delivery pipeline outage, relevance/algorithm change making notifications ignorable, OS-level notification permission change, over-notification causing mass mutes.
- **Product causes:** slow startup or crash on launch in the new version (users give up before the app opens), login loop bug.
- **External:** seasonal, competitor moment, OS update breaking something.
- Separate "fewer notifications sent" from "same notifications, fewer opens" - they have completely different owners and fixes.

62. Drivers are dropping out of a city on Lyft. How do you figure out what's going on?

Supply RCA

Same family as Q43 - go straight to driver economics and local conditions.

- **Quantify:** churn rate trend, which driver cohorts leave (new vs veteran, full-time vs part-time), hours-online before exit.
- **Earnings check:** earnings per hour trend vs other cities and vs competitor estimates; recent rate-card or bonus changes.
- **Demand check:** rides per online hour - if supply outgrew demand, everyone's earnings diluted.
- **Direct evidence:** exit surveys, support tickets, driver community forums, competitor recruiting offers in that city.
- **Local factors:** regulation, fuel cost, airport-policy change, safety incidents. Fix the economics, not the messaging; measure 30-day driver retention recovery.

63. B2C ad-based product: one day no ads are served and revenue drops to zero. Strategy?

Incident response

This is an incident, not a strategy question - first restore revenue, then prevent recurrence.

- **Triage (first hour):** is it demand-side (ad server/exchange down, account suspended, payment/billing issue with the ad network), or supply-side (ad SDK broken after a release, ad placements not rendering)?
- **Immediate:** roll back any suspect release, failover to a backup ad network, contact the ad partner's support with severity escalation.
- **Communicate:** internal stakeholders with impact estimate (revenue per hour lost); users usually don't need messaging unless experience broke.
- **After recovery - postmortem:** monitoring gap (why did we learn from revenue reports, not alerts?), single-point-of-failure (one ad network = risk), add fill-rate alerts and a second demand source.
- Also examine policy risk: if the account was suspended for policy violations, fix compliance before revenue resumes.

64. Your new feature boosts Amazon Search by 10% but adds 2 seconds to load time. What do you do?

Trade-off call

A 2-second slowdown is huge in e-commerce - don't ship blind; quantify the net effect.

- **Known evidence:** Amazon's own famous finding: ~100ms of added latency cost ~1% of sales; 2 seconds is an order of magnitude worse than that. Default assumption: net negative.
- **Verify with an experiment:** A/B the feature measuring not just search usage but downstream revenue per session and conversion - the 10% "search boost" must beat the latency tax on the full funnel.
- **Engineering options first:** async/lazy-load the feature, cache, precompute, load below the fold - often you keep most of the gain at a fraction of the cost.
- **Segment:** if value concentrates in one user type or device, ship it only there.
- Decision rule: ship only if revenue per session in the experiment is positive at 95% confidence; otherwise iterate on performance.

65. A metric for a video streaming service dropped by 80%. What do you do?

Root-cause funnel

An 80% collapse is almost certainly instrumentation or a catastrophic bug - check measurement before anything else.

- **Step 1:** which metric exactly, and when did it start? An instantaneous cliff = deploy or logging change; a fast slope = real behavior.
- **Step 2:** cross-validate with independent data (CDN bandwidth, server play requests, revenue) - if those are flat, the metric is broken, not the business.
- **Step 3 (if real):** segment hard - platform, region, content; an 80% real drop suggests playback failure (DRM/license server outage, CDN failure in a region, app crash on play).
- **Step 4:** incident-manage: roll back / failover, communicate, postmortem with monitoring gaps fixed.
- Rule of thumb: the bigger the drop, the more likely it's measurement or infrastructure, not user preference.

Section 3 - Estimation (Fermi) (Q66-Q87) - method: state assumptions, build the chain, round numbers, sanity-check

Interviewers score the structure, not the final digit. Say every assumption out loud, keep numbers round, and end with "does this feel sane?" All figures below are stated assumptions, not verified facts.

66. Estimate the number of tube lights used in a big city (10 million population). Fermi

Split usage into homes and non-homes (offices, shops, public spaces).

- **Homes:** 10M people / 4 per household = 2.5M homes. Assume 4 tube lights per home (kitchen, hall, bedrooms) = 10M.
- **Workplaces/shops:** assume 3M workers, 1 tube light per 2 workers (offices have many; small shops few, average it) = 1.5M; shops + restaurants, assume 500k establishments x 6 = 3M.
- **Public/institutions:** schools, hospitals, stations, streets - assume another 2M.
- **Total: roughly 15-17 million tube lights.** Sanity check: ~1.5 tube lights per resident feels plausible for a dense big city.
- State clearly: the multiplier assumptions (4/home, 6/shop) are the numbers to defend; the rest is arithmetic.

67. Estimate the cost of storing Google Photos. Fermi (cost)

Users x photos x size x storage price, with redundancy overhead.

- **Users:** assume 1B users with backup on. Average stored per user: assume 2,000 photos+ videos equivalent at ~4MB average (video-heavy users pull the average up) = 8GB per user.
- **Total storage:** 1B x 8GB = 8 exabytes. Add replication (data is stored ~2-3x for durability) = ~20EB effective.
- **Cost:** cloud storage runs roughly \$10-20 per TB per year at scale (their own infrastructure, cheaper). 20EB = 20M TB x ~\$15 = ~\$300M per year.
- **Answer: on the order of a few hundred million dollars per year.**
- Key sensitivities: photos-per-user and cost-per-TB - show which assumption moves the answer most.

68. How much time do people spend waiting in vehicles at traffic signals in the US?

Fermi (time)

Daily wait per driver x number of drivers x days.

- **Drivers:** ~230M licensed drivers in the US; assume 150M drive on a typical day.
- **Wait per driver:** urban/suburban trip passes ~10 signals; assume 30% turn red with ~45s average wait = $10 \times 0.3 \times 45\text{s} = \sim 135\text{s}$; round to ~2.5 minutes per trip, 2 trips a day = ~5 minutes per driver per day.
- **Daily total:** $150\text{M} \times 5\text{ min} = 750\text{M minutes} = \sim 12.5\text{M hours per day}$.
- **Yearly: ~4.5 billion person-hours.**
- Sanity check: 5 min/day of red-light waiting matches most commuters' intuition - good.

69. Estimate the daily bandwidth consumed by YouTube globally.

Fermi (bandwidth)

Watch hours x bitrate.

- **Watch time:** assume 2.5B monthly users, 500M daily viewers, 40 minutes average per day = ~330M watch-hours daily.
- **Bitrate:** blended quality (mobile 480p to TV 4K) - assume ~2 Mbps average = 0.9 GB per hour ($2\text{ Mbps} \times 3600\text{s} / 8 = 900\text{MB}$).
- **Daily bandwidth:** $330\text{M hours} \times 0.9\text{ GB} = \sim 300\text{ petabytes per day}$.
- **Answer: on the order of a few hundred PB/day.** Sanity check: YouTube is widely cited as a large double-digit share of global internet traffic - order of magnitude holds.
- Sensitivity: average bitrate dominates - 4K adoption changes everything.

70. How many people are watching YouTube in India right now?

Fermi (concurrent)

Daily viewers x fraction of the day they watch, adjusted for time of day.

- **Daily viewers:** assume 450M internet users in India, ~60% use YouTube daily = 270M daily viewers.
- **Concurrency:** average 30 min/day of viewing spread over ~16 waking hours = average concurrent = $270\text{M} \times (30/60)/16 = \sim 8.4\text{M}$.
- **Peak adjustment:** evening peak runs 3-4x the average; "right now" in the evening = ~25-35M concurrent viewers; mid-day, closer to 8-10M.
- **Answer: roughly 10M average, up to ~30M at evening peak.**
- Show the formula clearly: concurrent = daily users x daily minutes / minutes in waking day.

71. How many pianists are there in New York City?

Fermi

The classic: population x piano households x players per piano.

- **Households:** NYC ~8.5M people / ~2.5 per household = ~3.4M households.
- **Piano ownership:** assume 1 in 10 households has a piano = 340k pianos (keyboards included, state the inclusion).
- **Players:** assume average 1 person plays per piano-owning household = ~340k casual players.
- **Skilled/performing:** assume 3-5% play at a serious level = ~10-15k serious pianists.
- **Answer: ~300k+ people who can play, ~10k+ serious pianists.** The interviewer cares that you separated "can play" from "pianist".

72. How many queries per second does Gmail get? Fermi (QPS)

Users x actions per user, converted to a per-second rate.

- **Users:** assume ~1.8B Gmail users, ~40% active daily = ~700M DAU.
- **Actions:** each active user triggers ~20 requests/day on average (open, search, read, send, auto-sync/refresh counts too - background sync dominates).
- **Total:** $700M \times 20 = 14B$ requests/day. Per second: $14B / 86,400 = \sim 160k$ QPS average.
- **Peak:** peaks run ~2x average = ~300-350k QPS.
- **Answer: on the order of 10^5 requests per second.** Note how background sync, not human clicks, sets the rate.

73. Estimate total storage capacity for all videos on YouTube. Fermi (storage)

Upload rate x years accumulated x size per minute.

- **Upload rate:** commonly cited assumption ~500 hours uploaded per minute (state it as an assumption). Per year: $500 \times 60 \times 24 \times 365 = \sim 260M$ hours/year.
- **Accumulated:** assume average over ~18 years at a ramping rate $\approx$ equivalent of 8-10 years at today's rate = ~2-2.5B hours of video.
- **Size:** ~1GB per hour at standard quality, but YouTube stores multiple resolutions (~3-5x) = ~4GB/hour effective.
- **Total:** $2.2B \text{ hours} \times 4GB = \sim 9$ exabytes; with replication $\times 2-3 = \sim 20-30$ exabytes.
- **Answer: tens of exabytes including redundancy.** The multi-resolution and replication factors are what interviewers listen for.

74. Estimate Swiggy orders delivered per hour in India. Fermi (rate)

Users x frequency, then divide into hours.

- **Customer base:** assume 15M monthly active customers, ordering ~3 times/month = 45M orders/month = ~1.5M orders/day.
- **Distribution:** food delivery peaks at lunch (12-2) and dinner (7-11); assume 70% of orders land in ~6 peak hours.
- **Peak rate:** $1.5M \times 0.7 / 6 = \sim 175k$ orders/hour at peak; off-peak ~50k/hour.
- **Answer: ~100-200k orders/hour at peak, ~50k off-peak.**
- Cross-check from the supply side: assume 250k delivery partners doing 2.5 deliveries/hour at peak = ~600k capacity - demand within capacity, plausible.

75. Estimate passengers arriving at SF airport every day. Fermi

Flights x seats x load factor.

- **Flights:** SFO is a major hub; assume ~1,000 total flight movements/day, so ~500 arrivals/day.
- **Per flight:** mix of regional jets (~70 seats) and widebodies (~300); average ~150 seats, at ~80% load factor = ~120 passengers per arrival.
- **Daily arrivals:** $500 \times 120 = \sim 60,000$ passengers/day.
- **Answer: roughly 50-70k arriving passengers per day.** Sanity check: SFO serves ~50M passengers/year total = ~135k/day both directions; arrivals $\approx$ half = ~65k. Consistent.
- Showing the sanity-check against annual traffic is what separates a strong answer.

76. Estimate the number of WhatsApp chats occurring in India. Fermi (volume)

Users x messages per user per day.

- **Users:** assume ~500M WhatsApp users in India, ~80% daily active = 400M DAU.
- **Messages per user:** personal chats + groups; assume 30-40 messages sent per active user per day (group lurkers receive many, send few - count sends).
- **Daily messages sent:** $400M \times 35 = \sim 14$ billion messages/day.
- **If "chats" means conversations:** assume each user touches ~8 conversations/day = 3.2B active conversations/day.
- **Answer: ~10-15B messages/day, ~3B active conversations/day.** Clarify the unit ("chats" = messages or conversations?) before estimating - that clarification itself scores points.

77. How much storage for all images on Google Maps? Fermi (storage)

Decide what counts: user-contributed photos of places + Street View + satellite tiles. Here: place photos + satellite.

- **Place photos:** assume 200M places/businesses mapped, average 20 photos each at ~3MB = $200M \times 20 \times 3MB = \sim 12PB$.
- **Satellite imagery:** Earth's land area ~150M km²; at ~0.5m resolution compressed, assume ~0.5MB per km² tile-equivalent = ~75TB; multiple zoom layers and historical versions push it to a few PB.
- **Street View (if included):** see Q84 - tens of PB.
- **Total (place photos + satellite): order of 10-20PB raw, 30-50PB with replication.**
- The main move: decompose by image type - they have very different scales.

78. Estimate the total number of videos on YouTube. Fermi (count)

Upload rate x time, ramp-adjusted.

- **Rate today:** assume 500 hours/minute uploaded; average video ~10 minutes = ~3,000 videos/minute = ~4.3M videos/day.
- **History:** YouTube ran ~18 years with ramping uploads; effective full-rate years $\approx$ 8-10. $4.3M/day \times 365 \times 9 = \sim 14B$ videos.
- **Answer: on the order of 10 billion videos.**
- Cross-check from storage (Q73): 10B videos x 10 min = 1.7B hours - matches the storage estimate's hour count. Estimates that agree with each other impress interviewers.

79. Estimate total cars on Highway 101 between SF and Mountain View right now. Fermi (flow)

Cars on a road = flow per hour x hours of travel time (Little's Law).

- **Segment:** SF to Mountain View $\approx$ 40 miles, $\sim$ 30-45 min drive in traffic.
- **Flow:** a busy freeway lane carries $\sim$ 1,500-2,000 cars/hour; assume 4 lanes each direction = $\sim$ 8 lanes total, at $\sim$ 1,200 cars/hour/lane average = $\sim$ 9,600 cars/hour entering the corridor (both directions).
- **Occupancy:** travel time $\sim$ 0.75 hour, so cars on the road = $9,600 \times 0.75 = \sim 7,200$.
- **Answer: roughly 5,000-10,000 cars at any moment** (rush hour higher, night much lower - state the time dependency).
- Alternative density check: 40 miles $\times$ 8 lanes = 320 lane-miles; at $\sim$ 25 cars/mile/lane in moderate traffic = 8,000. Two methods agreeing = strong answer.

80. How many passengers are in planes in the air over the USA at any given time? Fermi

Daily passengers x fraction of the day spent airborne.

- **Daily passengers:** US airlines carry roughly 2.5-3M passengers/day (state as assumption).
- **Air time:** average flight $\sim$ 2.5 hours; passenger air-hours per day = $2.75\text{M} \times 2.5 = \sim 7\text{M}$ hours.
- **Concurrency:** flying is concentrated in $\sim$ 16 active hours, so average in the air = $7\text{M} / 16 = \sim 430\text{k}$.
- **Peak vs night:** midday peak $\sim$ 500-600k; at 3 AM, maybe 30-50k.
- **Answer: on the order of half a million people airborne during the day.**

81. Storage space for all Google Street View images? Fermi (storage)

Road distance x capture density x panorama size.

- **Coverage:** assume Street View covers $\sim$ 15M km of roads worldwide (state as assumption; published figures are in this range).
- **Capture density:** a 360-degree panorama every $\sim$ 10 meters = 100 panos per km = 1.5B panoramas.
- **Size:** each panorama at multiple zoom levels $\approx$ $\sim$ 5MB compressed = $1.5\text{B} \times 5\text{MB} = \sim 7.5\text{PB}$.
- **Historical versions + replication:** re-captures over years ($\sim$ 2x) and storage redundancy ($\sim$ 2x) = $\sim$ 30PB.
- **Answer: tens of petabytes.** Name the multipliers (versions, redundancy) - they're the differentiators.

82. Estimate the number of Tesla Model 3s in California. Fermi

Bottom-up from sales: total sold x California share x survival rate.

- **Global/US sales:** Model 3 launched 2017; assume $\sim$ 2M sold cumulatively, $\sim$ 50% in the US = $\sim$ 1M in the US.
- **California share:** California has historically bought an outsized share of US EVs - assume $\sim$ 35-40% = $\sim$ 350-400k.
- **Survival:** assume 95% still registered (crashes/exports) = $\sim$ 350k.
- **Answer: on the order of 300-400k Model 3s in California.**
- Cross-check top-down: $\sim$ 15M registered vehicles in CA; EVs $\sim$ 5-6% = $\sim$ 800k EVs; Model 3 as $\sim$ 40% of EVs = $\sim$ 320k. Consistent.

83. Estimate bicycles needed to start a bike-sharing operation in a big city. Fermi (ops)

Size the fleet from target rides and bike utilization.

- **Demand:** city of 5M; assume 5% try bike-share monthly = 250k users; assume 4 rides/user/month = 1M rides/month = ~33k rides/day.
- **Utilization:** a shared bike does ~4-6 rides/day in a healthy system (world benchmarks); assume 5.
- **Fleet:** $33k / 5 = \sim 6,600$ bikes in service; add ~15% for repairs/rebalancing = ~7,500 bikes.
- **Answer: roughly 7,000-8,000 bikes for launch.**
- Also mention: dock/parking density and rebalancing vans matter as much as fleet size - a good PM answer goes past the number to operations.

84. Estimate the number of Uber rides (per day, globally). Fermi (rate)

Riders x frequency, sanity-checked by drivers.

- **Demand side:** assume ~150M monthly riders globally, ~5 rides/month average = 750M rides/month = ~25M rides/day.
- **Supply check:** assume ~5M active drivers doing ~5 trips/day = 25M trips/day. Both sides agree - that's the point of the cross-check.
- **Answer: on the order of 20-30M rides/day globally.**
- Note the assumptions to defend: monthly riders and trips per driver are the swing factors; frequency per rider is the hardest to know.

85. How do you decide how many cash registers a new Walmart store needs? Fermi (capacity)

Registers = peak demand x service time, plus a wait-time target.

- **Peak demand:** assume the store serves ~10k customers/day, with ~30% in the peak 2 hours = 1,500 customers/hour at peak.
- **Service rate:** one cashier serves ~20 customers/hour (3 min per checkout).
- **Base need:** $1,500 / 20 = 75$ registers at peak... too many - adjust: large stores do run 25-30 lanes; refine peak to 800/hour (spread across more hours + self-checkout absorbing ~50%) = $800 / 20 = 40$, with self-checkout handling half = ~20 staffed lanes + 20 self-checkout kiosks.
- **Answer: ~20-30 staffed registers plus self-checkout**, sized to keep peak wait under ~5 minutes.
- The method is the answer: demand curve - service rate - wait-time SLA - technology mix.

86. How many dentists are there in New York? Fermi

Population needs / capacity per dentist.

- **Demand:** NYC ~8.5M people; assume 60% visit a dentist yearly, twice on average = ~10M visits/year.
- **Capacity:** one dentist handles ~15 patients/day x 220 working days = ~3,300 visits/year.
- **Dentists:** $10M / 3,300 = \sim 3,000$ practicing dentists.
- **Answer: roughly 3,000-4,000 dentists** (add specialists and it's a bit higher).
- Ratio sanity check: ~1 dentist per 2,500 residents - matches commonly cited US ratios. Always close with the ratio check.

87. Calculate the number of queries answered by Google per second. **Fermi (QPS)**

Users x searches per user per day / seconds.

- **Users:** assume ~4B internet users, ~85% use Google, ~70% daily = ~2.4B daily searchers.
- **Searches per day:** assume 3-4 searches per daily user = ~8.5B searches/day (public estimates cite a similar figure - fine to mention you recall the range).
- **Per second:** $8.5B / 86,400 = \sim 100,000$ queries/second average; peak $\sim 2x = \sim 200k$ QPS.
- **Answer: ~100k per second.**
- Interviewer bonus: mention autocomplete multiplies backend lookups several-fold if they ask about infrastructure.

Section 4 - Product Design (Q88-Q133) - framework: CIRCLES (Comprehend, Identify user, Report needs, Cut, List solutions, Evaluate, Summarize)

Pattern: clarify the goal and constraints first, pick ONE user segment and its top pain, then propose 2-3 solutions, evaluate, and land on a recommendation with a success metric.

88. Design a system to predict if something can go wrong in a cab ride. **CIRCLES + ML**

Goal: warn/intervene before safety or experience failures (route deviation, driver risk, breakdown).

- **Signals:** real-time - route deviation vs expected path, unexpected long stops, speed anomalies, time-of-day/area risk, driver ratings history, vehicle telematics (hard braking, maintenance codes).
- **Model:** a risk score per ride (classification on historical incident data); tiered thresholds - low: log only; medium: in-app check-in ("Everything OK?"); high: safety-team alert + share-ride prompt.
- **User experience:** silent by default; surface only when actionable (rider SOS, live trip sharing, support call-back).
- **False positives are the cost:** too many false alerts train users to ignore them - tune precision on the high tier.
- Metrics: incident rate per 10k rides, precision/recall of alerts, intervention response time.

89. Design a product for improving teenage health. **CIRCLES**

Teens ignore "health products" - the product must live inside habits they already have.

- **User & pain:** teens 13-17; top problems: poor sleep, junk food, inactivity, mental health stigma. Pick ONE: sleep (measurable, less stigmatized).
- **Solution:** a habit app with phone-based sleep tracking, a friend-league (teams compete on sleep streaks - social motivation beats lectures), and rewards that matter to teens (music subscriptions, game skins via sponsors).
- **Safeguards:** parental consent flow, no body-image content, no public shaming mechanics, teen-data privacy rules (COPPA-style).
- **Why it works:** uses peer motivation rather than authority; tiny daily commitment.
- Metrics: weekly active teens, average sleep-duration change at 4 weeks, streak retention, guardrail: screen-time at bedtime must fall, not rise.

90. Redesign a gas station (petrol pump). CIRCLES (physical)

The pain is dead time and distrust; EV transition is the strategic backdrop.

- **Pains today:** queue time, payment friction, mistrust of meter accuracy, nothing to do during the wait.
- **Redesign:** app-based pre-booking of a slot + automatic number-plate recognition for payment (drive in, fuel, drive out); transparent digital meter readout mirrored to your phone.
- **Revenue expansion:** convenience retail with order-ahead pickup at the pump, EV fast-charging bays with lounge, fleet/cab loyalty programs.
- **Layout:** one-way flow, separate EV and fuel zones, clear digital queue boards.
- **Metrics:** average service time per vehicle, throughput per hour, non-fuel revenue per visit, NPS on trust ("meter was accurate").

91. Design a meditation app for kids. CIRCLES

Kids won't sit still for adult meditation - make it a game; parents are the buyers.

- **Two users:** child (the user) and parent (the customer) - design for both.
- **Child experience:** 3-5 minute story-based sessions ("breathe with the dragon"), character progression tied to practice streaks, voice-guided with playful audio, no text-heavy UI.
- **Parent experience:** choose goals (sleep, focus, calm-down moments), see progress, co-do sessions.
- **Safety/ethics:** no ads, no streak-punishment anxiety, screen-time friendly (audio-only mode), COPPA compliance.
- **Metrics:** sessions per week per child, 4-week retention, parent renewal rate, bedtime-session completion.

92. Design a TV for a car. CIRCLES (hardware)

In-car viewing is for passengers (kids on trips, ride-share riders); safety constraints dominate.

- **Users:** back-seat families on long drives first; ride-share passengers second.
- **Design:** headrest-mounted or fold-down screen, individual wireless audio (no driver distraction), offline-downloaded content (connectivity is unreliable), parental controls and per-seat profiles.
- **Hard constraint:** nothing visible to the driver while moving; auto-dimming at night; motion-sickness mitigation (screen stability, horizon cues).
- **Content:** partnerships with streaming apps + download-before-trip flow; kids' profiles with time limits.
- **Metrics:** usage hours per trip, attach rate in family vehicles, content completion, driver-distraction incidents (must be zero).

93. Create a way to sell movie tickets on Facebook. CIRCLES (platform fit)

Movies are social plans - Facebook's edge is the friend graph, not the ticket inventory.

- **Insight:** people don't decide alone; the hard part is coordinating "who's in, which show".
- **Product:** movie pages with local showtimes, a "plan with friends" flow: propose 2-3 shows to a group - friends vote - one person books for all with split payment.
- **Integration:** partner with ticketing providers for inventory/payment (don't build theaters' backend); Events integration so the plan becomes a calendar event; reminders on show day.
- **Discovery:** "friends going to" signals in feed; trailer views convert to plan suggestions.
- Metrics: plans created, plan-to-purchase conversion, group booking share of tickets, revenue per movie page visit.

94. Design a mobile app to buy and sell used books. CIRCLES (marketplace)

Used books are low-value, high-friction items - the design must make listing nearly free.

- **Seller flow (the bottleneck):** scan ISBN barcode - auto-filled title/photo/condition - suggested price from recent sales - listed in under 30 seconds.
- **Buyer flow:** search + local pickup default (shipping kills thin margins), condition photos, buyer protection via in-app payment escrow.
- **Matching:** campus/city clusters (textbook seasons), wishlist alerts when a book lists nearby, bundle deals from one seller.
- **Trust:** ratings both sides, meetup-location suggestions (public places), dispute flow.
- Metrics: listing time (activation), listings per seller, sell-through rate in 30 days, repeat usage per semester cycle.

95. Design an app for an amusement park. CIRCLES

The park visit's biggest pain is queues and logistics; the app is a day-of companion.

- **Core features:** live wait times per ride, virtual queuing (reserve a slot, roam free), interactive map with routing, food order-ahead.
- **Trip planning:** build an itinerary before arrival (must-do rides, kid height filters), smart replanning when a ride breaks down.
- **Revenue features:** fast-pass upsells at decision moments, photo-pass integration, in-park offers.
- **Edge cases:** low battery (lite mode), poor signal (offline map), groups splitting up (location sharing opt-in).
- Metrics: rides per visitor per day (queue reduction = more rides), food/merch spend per visitor, NPS, app adoption rate at gate.

96. Design an app for making someone a better cook. CIRCLES

People don't want recipes, they want outcomes - cook tonight, with what's in the fridge, without failing.

- **Core loop:** photo/scan of ingredients - suggests 3 cookable recipes - hands-free guided cooking (voice step-by-step, auto timers).
- **Skill building:** technique micro-lessons embedded at the moment of use ("now dice the onion - 40s video"), difficulty progression, weekly skill path.
- **Feedback:** photo of the result - simple critique/score; substitution help when an ingredient is missing.
- **Retention hooks:** weekly meal plan, grocery list generation, household sharing (who cooks tonight).
- **Metrics:** cooked-meals per user per week (the real North Star), recipe completion rate, week-4 retention, grocery-list-to-cook conversion.

97. How would you design Google Maps for kids? CIRCLES (special users)

Kids need independence with guardrails; parents need peace of mind - design for the pair.

- **Kid experience:** landmark-based directions ("turn at the bakery"), big arrow UI, audio guidance, few safe destinations (home, school, grandma's) set by parents.
- **Safety layer:** parent-set geofences with alerts on deviation, no stranger-data (hide nearby people/business reviews), check-in button on arrival.
- **Learning angle:** gradually teaches navigation skills - fewer hints as the child grows (scaffolded independence).
- **Privacy:** minimal data collection, COPPA-compliant, location visible only to the family group.
- **Metrics:** successful independent trips, parent alert rate (should be low), kid-initiated usage, family retention.

98. Design a cafeteria app for employees. CIRCLES (B2B2C)

Workday lunch pain: the 12:30 rush, wasted waiting, and decision fatigue.

- **Core:** daily menu with photos, pre-order + time-slot pickup (skip the line), wallet/payroll-deducted payment.
- **Smart bits:** "usual order" one-tap reorder, dietary filters (veg/allergen), group orders for teams, live queue status.
- **For the cafeteria operator (the real customer):** demand forecasting (cook to pre-orders = less waste), capacity smoothing via slot pricing, feedback per dish.
- **Rollout:** pilot one building, measure line-time reduction before scaling.
- **Metrics:** pre-order share of meals, average wait time, food waste %, weekly active employees, operator margin improvement.

99. Create a playlist for a user after they listen to a single song (like Spotify). Algorithm design

One song is a weak signal - expand it using similarity, then refine with every skip/save.

- **Seed expansion:** content similarity (tempo, genre, era, mood from audio features) + collaborative filtering ("people who loved this also loved...").
- **Ranking:** blend familiarity (some known tracks = trust) with discovery (2-3 new artists); order by energy arc, not random.
- **Online learning:** every skip within 30s downweights similar tracks; saves/repeats upweight; the playlist reshapes within the session.
- **Cold start edge:** if the user is new, lean on the seed song's global co-play data; if known, bias to their taste profile.
- Metrics: % of playlist played without skip, save rate, session extension (did they keep listening after the playlist ended?).

100. Design WhatsApp for children. CIRCLES (safety-first)

Kids need family connection without open-network risk; every feature passes through parental control.

- **Contact model:** parent-approved contacts only - no discoverability, no phone-number sharing, requests go to the parent.
- **Features:** chat, voice/video calls with approved contacts, family groups; fun layer (stickers, drawing) since that's why kids want it.
- **Parent dashboard:** contact approvals, screen-time windows, alerts on flagged content (bullying language detection), NOT message reading by default (trust vs surveillance trade-off - state it).
- **Compliance & safety:** COPPA-style consent, no ads, no public groups/channels, reporting flows a child can actually use.
- Metrics: weekly active kids, parent approval completion rate, flagged-content precision, family-group activity. Guardrail: zero contact from unapproved accounts.

101. You're a new PM at Facebook. Design an app to help people save money. CIRCLES

Saving fails from friction and absent motivation - automate the action, socialize the motivation.

- **Core mechanic:** automatic round-ups and scheduled micro-transfers to a goal pot (link bank account/card); "save \$5 when you skip a coffee" nudges.
- **Facebook's edge:** group goals - friends saving together for a trip; visible progress to a private circle creates accountability.
- **Behavior design:** celebrate milestones, never shame; visualize "future you" outcomes (goal date, what it buys).
- **Trust:** partner with regulated banks for holding funds; fees transparent; withdrawal any time (liquidity fear kills saving apps).
- Metrics: % of users with an active auto-save rule after 30 days, goal completion rate, average balance growth, 90-day retention.

102. Design an ATM for airports. CIRCLES (context design)

Airport users are rushed, foreign, and security-conscious - the ATM must serve a stressed traveler in 30 seconds.

- **User needs:** foreign travelers needing local currency, multi-language, clear fees, card safety anxiety.
- **Design:** language auto-detect from inserted card + icon-based language selection; up-front fee and exchange-rate transparency; cardless withdrawal via QR from banking app (card skimming fear); e-receipt to email.
- **Context features:** dispensing mixed denominations (travelers need small bills for taxis/tips), located pre-security AND at arrivals, queue-time display.
- **Edge cases:** card swallowed flows with on-site support, cash recycling uptime SLA (an empty airport ATM is a broken promise).
- **Metrics:** transaction completion rate, time per transaction, international-card usage share, uptime, fee-complaint rate.

103. Design an alarm clock for the blind. CIRCLES (accessibility)

Every interaction that assumes sight must be redesigned: setup, time-check, dismiss, snooze.

- **Interaction:** voice-first setup and control ("set alarm 6:30"), tactile large buttons with distinct shapes, audio/haptic feedback for every action (confirmations spoken).
- **Wake experience:** escalating alarm - gentle vibration pad under pillow + sound; voice time announcement on wake ("6:30, Thursday").
- **Differentiation:** smart-home integration (lights, kettle), news/weather briefing option, battery backup with spoken low-battery warning.
- **Test with users:** co-design with blind users from day one - assumptions about tactile UI are often wrong.
- **Metrics:** task success rate for set/change/cancel without assistance, wake reliability, returns/support tickets, NPS in the blind community.

104. Build a Google product that enables people to learn new skills. CIRCLES

Learning fails at motivation and practice, not content access - design for finishing, not starting.

- **Positioning:** Google's edge is search intent + YouTube how-to content - turn "how do I..." searches into structured learning paths.
- **Product:** skill paths (e.g. "guitar basics in 30 days") assembling the best YouTube lessons + daily 15-minute practice tasks + AI feedback (record yourself, get tips).
- **Retention mechanics:** streaks with forgiveness, practice reminders, proof-of-skill artifacts (shareable clips/badges), community of co-learners per cohort.
- **Monetization (optional):** certification for job skills, Workspace/Classroom tie-ins.
- **Metrics:** path completion rate (the industry's weak point), minutes practiced per week, skill-verification pass rate, 8-week retention.

105. Design a bookshelf for kids. CIRCLES (physical)

Kids choose books by cover, not spine - and the shelf must survive the kid.

- **Users:** children 2-8, with parents as buyers; needs differ: toddler (safety, access) vs early reader (choice, independence).
- **Design:** front-facing display shelves (covers visible) at child height; rounded edges, tip-proof anchoring, wipeable material.
- **Growth features:** adjustable shelf heights, a "rotation" top row parents use to feature this week's books (novelty drives reading), book-return bin to make cleanup a game.
- **Delight:** reading-nook integration (seat + light), theme colors.
- Success measures: kid-initiated book pickups per day (observed in user testing), parent repeat purchase/recommendation, safety record.

106. Design a fridge for a person in a wheelchair. CIRCLES (accessibility)

Standard fridges fail on reach height, door force, and deep-shelf access.

- **Constraints:** seated eye level ~1.1m, limited reach depth, one-hand operation often needed, side approach to the appliance.
- **Design:** bottom-drawer heavy layout (most-used items 0.6-1.2m height), motorized pull-down upper shelves, side-by-side or French-door with low-force handles and auto-open assist, pull-out trays instead of deep shelves.
- **Smart layer:** voice control, interior camera viewable on phone (see the back without reaching), open-door alerts.
- **Kitchen fit:** works with lowered counters and knee-clearance layouts; test with wheelchair users in real kitchens.
- Metrics: task success rate for retrieve/store without assistance, force required per interaction, user testing satisfaction.

107. Design a referral system for a grocery app. Growth design

Grocery is habitual and margin-thin - referrals must create habits for both sides without burning margin.

- **Structure:** two-sided reward: referrer gets credit when the friend completes a SECOND order (not first - filters coupon hunters), friend gets a first-order discount.
- **Timing:** prompt referral after a delight moment (first on-time delivery, savings milestone), not at signup.
- **Mechanics:** personal referral code + one-tap share to WhatsApp; group/household invites (families order together); tiered rewards for top referrers.
- **Fraud controls:** device/payment-method fingerprinting, cap rewards, delay credit until return-window passes.
- Metrics: K-factor (invites per user x conversion), CAC of referred vs paid users, 60-day retention of referred customers, reward cost as % of order margin.

108. Design a Smart Refrigerator for the USA (Google entering appliances). **CIRCLES + strategy**

Don't start with a touchscreen on the door - start with jobs a fridge can uniquely do.

- **Core jobs:** know what's inside (inventory), reduce waste (expiry alerts), simplify restock (auto grocery lists, one-tap reorder), meal suggestions from actual contents.
- **Tech:** interior cameras + weight sensors, Google Assistant integration, family profiles; phone app does heavy lifting (screen optional, cost down).
- **Ecosystem play (why Google):** ties Assistant, Google Shopping, calendar ("dinner guests Saturday - check ingredients"), energy optimization with Nest.
- **Market entry:** partner with an established manufacturer first (brand trust in appliances matters), premium tier first.
- **Metrics:** food-waste reduction per household (survey + sensor), restock-list usage, Assistant interactions per week, attachment rate of reorder flows.

109. Create a smartphone for a 6-year-old. **CIRCLES (safety-first)**

The product is really "parent-managed communication device" - the child is the user, the parent is the customer.

- **Core:** calls/messages with parent-approved contacts only, GPS location for parents, SOS button, no open app store/browser.
- **Parent controls:** app approvals, screen-time schedules (school mode, bedtime lock), content filtering by age.
- **Kid appeal:** camera with fun filters (parent-visible gallery), a few curated games/educational apps, durable bright hardware with bumper case.
- **Hard requirements:** COPPA compliance, no ads, no in-app purchases without parent PIN, low-cost data plan.
- **Metrics:** parent setup completion, daily parent-child touchpoints, SOS false-alarm rate, device survival rate (drops!), family retention at upgrade cycle.

110. Design a Facebook tool for the Olympics. **CIRCLES (event product)**

The Olympics is a 2-week global shared moment - the tool should make following and celebrating it social.

- **Hub:** an Olympics destination - schedule in your timezone, live medal table, highlights, athlete pages.
- **Social layer:** country-based fan groups, friend reactions during live moments, "your friends are watching" prompts, watch-party integration.
- **Athlete side:** verified athlete stories/behind-the-scenes content (exclusive feel), Q&A sessions.
- **Constraints:** broadcast rights (highlights only where licensed - state this), spoiler controls (time-shifted viewers), moderation at global scale.
- **Metrics:** daily Olympics-hub visitors during the games, social interactions per event moment, group joins, post-Olympics retention of new follows/groups.

111. Design a washing machine for Google. CIRCLES

Laundry pain is forgetting, guessing settings, and timing - software solves all three.

- **Smart core:** auto-detect load (weight/fabric via sensors) - picks the cycle itself; detergent auto-dosing from a refillable tank with reorder integration.
- **App experience:** start/schedule from phone (finish when you get home), notifications when done (the #1 pain: forgotten wet clothes), energy-price-aware scheduling.
- **Maintenance:** predictive diagnostics (bearing wear from vibration patterns), self-clean reminders, service booking in-app.
- **Why Google:** Assistant voice control, Nest energy integration, retail data for detergent partnerships. Go via an appliance-manufacturer partnership, not own factories.
- **Metrics:** forgotten-load incidents, cycle-auto-select acceptance rate, energy per wash, app engagement, warranty claims.

112. Design a fridge for kids. CIRCLES (special users)

Interpret as a fridge experience designed around kids' safe independence (snacks) with parent control.

- **Needs:** kids 4-10 want snacks independently; parents want healthy choices and no mess/midnight raids.
- **Design:** a kid-zone drawer at child height parent-stocked with approved items; transparent panel or internal camera so kids see options without holding the door open (energy + mess).
- **Controls:** time-based locks (no post-9pm opening without parent PIN), portion compartments, spill-proof shelving, finger-safe gaskets and soft-close.
- **Fun/learning:** magnetic face showing snack "budget" for the day, healthy-choice gamification.
- **Metrics (concept test):** unsupervised healthy-snack success rate, door-open duration, parent satisfaction, safety incidents.

113. Design a driverless car pick-up service for kids. CIRCLES (trust design)

The real product is parental trust; the car is table stakes. Every feature answers "how do I know my child is safe?"

- **Core flow:** parent books recurring school runs; verified pickup (child + car authenticate via app/PIN/face), fixed known routes first (school-home).
- **Safety stack:** live video + GPS for parents, no other passengers (or only verified schoolmates), remote human operator monitoring every ride, geofenced routing with alerts.
- **Handover:** authenticated drop-off - car waits until the child enters the building/confirmed adult check-in.
- **Rollout:** geofenced suburb pilot, daylight hours, school partnerships; regulation and insurance are the gating items - name them.
- **Metrics:** rides without intervention, parent NPS, incident rate, remote-operator interventions per 100 rides (must trend to near zero).

114. Launch Uber in a new city. **Launch playbook**

A marketplace launch is a cold-start problem: seed supply, spark demand, reach liquidity, then sustain.

- **Pre-launch:** regulatory check, city selection criteria (population density, smartphone penetration, taxi pain, tourism), competitive scan.
- **Seed supply first:** driver recruitment with guaranteed earnings for early drivers (removes their risk), onboarding events, referral bonuses.
- **Spark demand:** launch promos (first rides free/discounted), partnerships (airport, hotels, events), local marketing; start in the densest zone, not the whole city.
- **Liquidity target:** rider ETA under 5 minutes in the core zone and driver utilization healthy - only then expand zones.
- Metrics: ETA and fulfillment rate (liquidity), driver earnings/hour, rider week-4 retention, CAC payback, regulatory standing.

115. Design Lyft for kids. **CIRCLES (trust design)**

Human-driven version of Q113: vetted drivers + full parent visibility + controlled handovers.

- **Driver vetting (the product):** enhanced background checks, childcare-experience preference, kid-safety training certification, small consistent driver pool so families get familiar faces.
- **Experience:** parent books and tracks live; child check-in via PIN; car seats provided by age; no ride changes without parent approval.
- **Handover protocol:** verified pickup (driver ID shown to child via app), drop-off confirmation with photo/geofence at destination.
- **Scope:** start with scheduled recurring rides (school, activities), not on-demand - predictability builds trust and supply quality.
- Metrics: on-time pickup, parent NPS, repeat booking rate, safety incident rate, driver retention in the program.

116. Flipkart wants to grow users above age 60. What will you do? **CIRCLES (segment growth)**

Senior e-commerce adoption fails on trust and usability, not interest - design for both.

- **Barriers:** small text/complex flows, payment anxiety, fear of fraud/returns hassle, language.
- **Product:** senior mode - large fonts, simplified 3-step checkout, regional-language voice search, prominent call-for-help (assisted shopping via phone), cash-on-delivery emphasis.
- **Trust:** clear return promise in plain language, fraud-awareness tips, verified-seller badges, family account linking (adult children can approve/help remotely).
- **Acquisition:** partner channels seniors trust - community centers, newspapers, TV; categories first: medicines-adjacent wellness, books, devotional, grocery staples.
- Metrics: 60+ registrations and activation (first order), checkout completion rate vs baseline, support-call rate, 90-day reorder rate in segment.

117. Design a vending machine for blind people. CIRCLES (accessibility)

Selection, payment, and collection all assume sight - redesign each step for audio/tactile use.

- **Interaction:** tactile keypad with braille + audio menu (headphone jack or Bluetooth to phone for private audio), voice confirmation of selection and price before payment.
- **Payment:** contactless tap (easier than cash), spoken confirmation of charge and change.
- **Collection:** standardized collection-bin position with tactile guide strip, audio "your item is ready" cue, jam detection with auto-refund (a blind user can't shake the machine).
- **Consistency matters:** same button layout across all machines so the skill transfers.
- Metrics: unassisted purchase success rate, time per transaction, error/refund rate, feedback from blind-user testing panels.

118. How would you design a gardening app? CIRCLES

Gardening failure is mostly forgetting and not-knowing - the app is a reminder engine plus a plant doctor.

- **Core:** add your plants (photo ID) - personalized care schedule (water/feed/prune reminders by species, season, local weather).
- **Diagnosis:** photo a sick leaf - likely cause + treatment (image recognition), with human-expert escalation as premium.
- **Planning:** what-to-plant-when calendar for your zone, companion-planting guides, garden layout planner.
- **Retention:** growth journal with photos (emotional investment), community per plant type, seasonal re-engagement (spring onboarding is the moment).
- Metrics: plants kept alive past 90 days (the honest North Star), reminder action rate, weekly active gardeners in season, premium conversion for diagnosis.

119. Design an app for the US market that helps patients find doctors. CIRCLES (healthcare)

US patients' real pains: insurance fit, availability, and opaque prices - a directory alone solves nothing.

- **Core filters:** in-network insurance check (enter plan - see only covered doctors), specialty + condition, next-available appointment, real patient reviews.
- **Booking:** real-time slots booked in-app (no phone tag), telehealth option, price estimate before booking.
- **Trust layer:** verified credentials, outcomes data where available, reviews only from verified visits.
- **Compliance (name it):** HIPAA for any health data; doctor-side onboarding is the supply bottleneck - start with one specialty/city.
- Metrics: search-to-booking conversion, appointment completion (no-show rate), % bookings with insurance pre-verified, repeat booking rate.

120. Build a product on Facebook for blood donation. CIRCLES (social good)

Blood donation is a supply-timing problem: the right donor, alerted at the right moment, near the need.

- **Donor side:** opt-in registry (blood type, city), notifications when a nearby hospital/blood bank needs your type - Facebook's reach and notifications are the edge.
- **Requester side:** hospitals and verified patients post needs; friends-of-friends propagation ("someone near you needs O-").
- **Habit layer:** donation history, reminders when eligible again (~3 months), badges and thank-you shares (social proof drives donation).
- **Trust & safety:** verify requesting institutions (prevent scams), no medical-data exposure beyond what the donor shares, connect to official blood banks rather than peer-to-peer payment.
- Metrics: registered donors, notification-to-donation conversion, fulfillment time per request, repeat-donation rate.

121. How would you design a "Google Refrigerator"? CIRCLES

Same core as Q108 - emphasize what Google uniquely adds beyond a smart appliance.

- **Differentiation:** Google AI as the brain - visual inventory recognition, meal planning from contents + your calendar + dietary goals, proactive "the milk expires tomorrow" via Assistant.
- **Key flows:** hands-free add-to-list while cooking, family-shared inventory (no more duplicate buying), recipe-to-cart with retail partners.
- **Ecosystem:** Android/Assistant/Nest integration; camera and sensors feed Google's AI advantage that pure appliance makers lack.
- **Go-to-market:** partner with a major manufacturer, premium SKU first, sell the software subscription (advanced planning/features) as the margin.
- Metrics: weekly app/Assistant interactions per household, grocery-waste reduction, list-to-purchase conversion, subscription attach rate.

122. Design Uber for kids. CIRCLES (trust design)

Same trust architecture as Lyft-for-kids (Q115): vetted drivers, live visibility, controlled handovers.

- **Driver pool:** separate "Uber Kids" certified tier - deeper background checks, child-safety training, consistent drivers per family.
- **Product:** scheduled recurring rides, guardian live-tracking with in-app video check option, PIN-based child pickup verification, car-seat options.
- **Handover:** drop-off confirmed only when the child checks in at destination (geofence + adult confirmation).
- **Launch:** one family-dense city, school partnerships, premium pricing (trust is worth paying for), liability/insurance framework first.
- Metrics: family repeat rate, on-time performance, incident rate, guardian NPS, driver program retention.

123. Design Netflix for senior citizens (goal: increase engagement time). **CIRCLES (segment)**

Seniors bounce off choice overload and fiddly UI - simplify ruthlessly and surface familiar content.

- **UI:** senior profile mode - large text and posters, high contrast, fewer rows, obvious "Continue Watching", simple remote-friendly navigation.
- **Content:** curated collections by era/genre they love (classics, devotional, regional language), human-curated "tonight's pick" instead of infinite scroll.
- **Accessibility:** clear audio mode (dialogue enhancement), large subtitles, playback-speed control.
- **Social:** watch-with-family remote sessions (grandkids recommend, one tap to play what they sent).
- Metrics: session length and sessions per week in the segment, time-to-first-play (should drop sharply), search abandonment, retention of 60+ subscribers.

124. Launch Google Maps in a new region. **Launch playbook**

Maps launch quality = data quality; the playbook is about acquiring local ground truth fast.

- **Data foundation:** base map from satellite + partner data, then local POI acquisition (business listings, government data, Local Guides community).
- **Localization:** local language + script, local addressing quirks (many countries don't use street addresses - landmarks matter), offline maps for low-connectivity areas.
- **Launch sequencing:** navigation core first (driving), then transit partnerships, then local discovery; recruit Local Guides early for reviews/photos.
- **Regulatory:** mapping data laws (some countries restrict map data - e.g. local-partner requirements); name this as a gating item.
- Metrics: POI coverage vs ground truth, navigation success rate, DAU growth, correction/report rate from users (quality proxy), Local Guide contributions.

125. Design a new location-sharing app for Google. **CIRCLES**

Location sharing is a trust product: people share safety-relevant location with specific people for specific times.

- **Use cases to prioritize:** family safety (kids, elderly parents), friend coordination ("are you close?"), trip safety (share until I arrive).
- **Design:** granular sharing - per person, per duration, with auto-expiry default (privacy by default); arrival/departure notifications instead of constant tracking.
- **Differentiators:** integrates with Maps (share trip progress natively), battery-efficient passive updates, works cross-platform via link (no app needed for viewer).
- **Privacy (the make-or-break):** visible indicator whenever shared, one-tap stop, no history kept by default, anti-stalking safeguards (periodic reminders that you're sharing).
- Metrics: weekly active sharers, session-based share completion (arrived - auto-off), notification usefulness rate, revocation-after-reminder rate (trust signal).

126. Design a car-sharing platform (like Uber) for disabled people. CIRCLES (accessibility)

The gaps are vehicle fit, driver training, and booking confidence - each becomes a feature.

- **Supply:** WAV (wheelchair-accessible vehicle) fleet with ramps/lifts, recruited and trained drivers (assistance certification), incentive structure so WAV supply is reliable.
- **Booking:** accessibility profile per rider (wheelchair type, assistance level, service animal) - matched to suitable vehicles; guaranteed-ride scheduling for medical appointments (reliability is the core promise).
- **Experience:** longer pickup windows without cancellation penalties, trained-driver badges, in-app communication for deaf riders (text-first), fare transparency with subsidy integration (government/insurance programs).
- **Launch:** partner with hospitals and disability organizations; one city, scheduled rides first.
- **Metrics:** fulfillment rate for WAV requests (the hard one), average wait, repeat usage, driver assistance-rating, complaint rate vs mainstream product.

127. Design a consumer app for a scooter-sharing business. CIRCLES (micromobility)

The app wins on find-unlock-ride friction and on keeping riders safe and legal.

- **Core flow:** map of nearby scooters with battery levels - reserve - QR unlock - ride - photo-verified park-to-end.
- **Safety/compliance:** first-ride tutorial, helmet prompt, geofenced no-ride/slow zones, parking rules with photo check (cities ban operators over sidewalk clutter - this is existential).
- **Pricing:** unlock fee + per-minute, day passes for commuters, low-income program (cities increasingly require it).
- **Ops feedback loop:** report-damage flow, battery-aware routing (don't strand riders).
- **Metrics:** rides per scooter per day (utilization), unlock-to-ride conversion, parking-compliance rate, safety incident rate, contribution margin per ride.

128. Design a library for the future. CIRCLES (vision)

When books are free digitally, the library's future is access to what money can't easily buy: space, tools, guidance, community.

- **Pillars:** (1) Space - quiet work, maker labs, studios (recording, 3D printing); (2) Access - digital lending, device/WiFi lending, subscriptions bundled; (3) Guidance - librarians as learning coaches, AI-assisted discovery; (4) Community - events, classes, local memory archives.
- **Digital layer:** one app for catalog + room booking + e-lending + event RSVP; personalized reading paths.
- **Equity mission:** free access to expensive tools and connectivity for those who can't buy them - the modern version of the original mission.
- **Funding reality:** public budget + memberships for premium spaces + partnerships.
- **Metrics:** visits and active members, digital checkouts, tool/space utilization, program attendance, demographic reach vs local population.

129. Design a supermarket for older people. CIRCLES (physical)

Design for reduced mobility, vision, and stamina - without making it feel like a hospital.

- **Layout:** rest seating every aisle, wide aisles for mobility aids, essentials at waist height (no deep bending/reaching), non-slip floors, bright even lighting.
- **Reading & decision aids:** large-font shelf labels and price tags, magnifiers on carts, simple store map, staff trained for unhurried assistance.
- **Service layer:** assisted shopping (staff or family remote list), carry-to-car, home delivery for heavy items, quieter "senior hours".
- **Checkout:** no-rush lanes, seated checkout option, help with bags; clear simple digital payment help or cash always accepted.
- **Metrics:** visit duration comfort (survey), assistance-request rate, repeat visit frequency, basket completion (did they get everything on their list?), injury incidents.

130. Build a product to solve the dog poop problem. CIRCLES (open brief)

Clarify whose problem: city cleanliness (unscooped poop in public) vs owner convenience. Pick public-space compliance.

- **Root causes:** owners forget bags, no bins nearby, no consequence for not scooping.
- **Product:** smart bin network - solar compacting bins with free bag dispensers at parks/trails; app map of bins and bag stations; gamified "clean walk" streaks and community reporting of hotspots.
- **Business model:** city contracts (cleanliness budgets) + pet-brand advertising on dispensers; data on hotspots helps cities target enforcement.
- **Alternatives considered:** DNA registration (expensive, hostile), fines alone (no infrastructure) - infrastructure + convenience beats punishment.
- **Metrics:** complaints to city, bin usage rates, hotspot reduction, cost per ton collected vs street sweeping.

131. Design a web search engine for children under 14. CIRCLES (safety-first)

Kids need real answers to real questions inside a walled, age-appropriate garden.

- **Content:** curated whitelist index + strict safe-search, results filtered by reading level, zero ads.
- **Experience:** visual results (thumbnails, videos), voice search (pre-readers!), question-style queries encouraged ("why is the sky blue"), instant-answer cards with simple language.
- **Safety & privacy:** no personal-data collection (COPPA), no tracking/profiling, parent dashboard for activity themes (not surveillance of every query - balance trust).
- **Learning layer:** "did you mean" spelling help, related-curiosity suggestions, source reliability hints (early media literacy).
- **Metrics:** successful-answer rate, reformulation rate, weekly returning kids, blocked-content appeal/overblock rate, parent satisfaction.

132. Design a bicycle-renting app for tourists. CIRCLES

Tourists don't know the city - the app sells confident exploration, not just bikes.

- **Core:** find/unlock/pay in any language and currency, no local phone number required, transparent per-hour/day pricing.
- **Tourist layer (the differentiator):** curated self-guided routes (highlights in 2 hours, food tour), audio guide triggered by GPS, offline maps.
- **Trust:** bike condition guarantee (report-and-swap), helmet availability, insurance/liability clarity upfront, 24/7 support chat.
- **Ops:** hotel/partner pickup points, flexible drop zones, family/kid options.
- **Metrics:** rentals per bike per day in season, route-completion rate, support tickets per 100 rentals, NPS from tourists, repeat/referral within the same trip group.

133. Build a product to buy and sell antiques. CIRCLES (marketplace)

Antiques are high-value, trust-scarce transactions - the product is authentication and confidence, not listings.

- **Trust core:** expert verification service (photo review, optional in-person appraisal), provenance documentation upload, escrow payments until delivery acceptance.
- **Marketplace:** rich listings (condition reports, restoration history, high-res photos/video), category specialists (furniture, art, coins), auctions + fixed price both supported.
- **Logistics:** insured specialty shipping partners for fragile/oversize items - a top reason such deals fail.
- **Two-sided launch:** seed with established dealers (they bring inventory and credibility), then attract collectors; take rate on verified transactions.
- **Metrics:** authenticated share of listings, transaction completion rate, dispute/return rate, average order value, dealer monthly sales, buyer repeat rate.

Section 5 - Product Improvement (Q134-Q169) - framework: clarify goal - pick segment & pain - solutions - prioritize (RICE) - measure

Pattern: "improve X" is still a design question. Anchor on a goal and ONE user segment, find their biggest pain, propose 2-3 fixes, prioritize with value/effort, name the metric that proves it worked.

134. Improve the Quora website and your strategy to engage more users. Improvement

Quora's core loop (good answer - more readers - more writers) weakened as answer quality diluted; fix quality signals.

- **Segment:** knowledge seekers who land from search and never return; pain = low-quality/SEO-bait answers burying good ones.
- **Improvements:** stronger answer ranking (expertise signals, credentials verification), better question deduplication, follow-topics onboarding to build a personalized feed in week 1.
- **Engagement:** writer side - distribution feedback (views/upvotes notifications), spaces/communities; reader side - daily digest of top answers in followed topics.
- **Prioritize:** answer-ranking quality first (RICE: reaches everyone, high impact on trust).
- **Metrics:** upvote rate per answer view, D30 retention of new readers, answers per writer per month, search-visitor return rate.

135. Redesign the Apple App Store. Improvement

The store is great for search-and-install, weak at discovery for users and fair visibility for developers.

- **Users' pain:** hard to find good apps beyond top charts; reviews gamed.
- **Improvements:** personalized discovery feed (based on usage patterns, not just installs), curated collections by real editors with "why we picked this", verified review weighting (usage-verified reviews rank higher), trial-before-buy as standard.
- **Developer side:** clearer ranking factors, better analytics, anti-copycat enforcement.
- **Trade-off to name:** discovery vs Apple's privacy stance (personalization must stay on-device).
- Metrics: % of installs from browse/discovery vs search, install-to-open rate, 30-day retention of discovered apps, developer satisfaction.

136. Improve Grammarly. Should Google acquire it? Improvement + M&A

Grammarly's value is cross-app writing assistance; its threat is platform owners bundling the same thing free.

- **Improvements:** deeper context (tone for audience, domain-specific style - legal, academic), team style guides for businesses, multilingual support, generative rewrite features with factual-integrity checks.
- **Acquisition logic for Google:** fits Docs/Gmail ecosystem, adds a paid productivity tier, brings writing-quality AI talent and data.
- **Against:** Google can build similar features into Workspace natively; antitrust scrutiny on Big Tech M&A; Grammarly's cross-platform neutrality is its moat and would weaken inside Google.
- **Recommendation structure:** acquire only if build-vs-buy shows 3+ year advantage and regulatory path is plausible; otherwise partner/integrate.
- Metrics to watch either way: enterprise seat growth, free-to-paid conversion.

137. Improve Duolingo. Improvement

Duolingo is great at habit, weaker at real fluency outcomes - improve the bridge from streak to speaking.

- **Segment:** motivated learners past month 3 whose progress feels plateaued.
- **Improvements:** AI conversation practice (voice role-plays with feedback), real-world content (podcasts/articles at graded difficulty), grammar explanations surfaced at the error moment, peer speaking clubs.
- **Monetization fit:** conversation practice as premium tier - aligns revenue with the highest-value feature.
- **Guardrail:** don't break the streak/gamification magic that drives retention.
- Metrics: speaking-practice adoption, CEFR-level progression of active users, D90 retention of engaged users, premium conversion.

138. You are the PM of YouTube Shorts. What improvements would you make? Improvement

Shorts' gaps vs TikTok: creator monetization and original-content culture.

- **Creator side:** clearer revenue share for Shorts, better creation tools (templates, effects, remix), analytics for short-form specifically.
- **Viewer side:** stronger interest graph training early (better "not interested" feedback), series/playlists for shorts so creators build followings not just views.
- **YouTube's unique edge to press:** funnel from Shorts to long-form and subscriptions - make the "watch the full video" path seamless.
- **Guardrails:** Shorts shouldn't cannibalize long-form watch time (the revenue engine); measure total-platform watch time, not Shorts alone.
- Metrics: Shorts DAU, creator earnings per 1k views, Shorts-to-long-form conversion, total platform watch time.

139. Improve LinkedIn Messenger. Improvement

LinkedIn messaging is professional but noisy - the pain is spam and low-context conversations.

- **Segment:** active networkers and job seekers buried under sales InMails.
- **Improvements:** smart inbox triage (connections vs recruiters vs sales - separate tabs), context cards (shared connections, recent interactions) on each thread, templated quick replies, scheduling integration (book a call without leaving chat).
- **Sender quality:** rate-limit cold outreach, response-rate-based sender reputation (spammy senders get less reach).
- **Trade-off:** InMail spam pays the bills - reduction must be balanced with recruiter revenue; state it.
- Metrics: response rate to first messages, % of conversations marked useful, unsubscribe/mute rate, recruiter conversion (guardrail).

140. How would you increase YouTube's daily active users? Growth

DAU = more users coming daily OR existing users coming more days - attack the bigger lever: visit frequency.

- **Diagnose first:** is the gap in acquisition (flat new users) or frequency (users visiting 3 days/week instead of 6)? Data decides the play.
- **Frequency levers:** subscriptions that matter (notify well, not often), daily-habit surfaces (Shorts feed, community posts from subscribed channels), streak-like "new from your channels" digest.
- **Acquisition levers:** emerging-market lite experience (data-light, offline), TV/living-room growth, Shorts as the top-of-funnel.
- **Guardrail:** notification-driven DAU that burns trust; watch opt-outs and uninstalls.
- Metrics: DAU/MAU, visits per user per week, notification-to-session rate with opt-out guardrail.

141. Improve Instagram Stories. Improvement

Stories are mature; the frontier is creation effort (too high for casual users) and meaningful interaction.

- **Creation friction:** AI-assisted story drafts (auto-layout from recent photos), templates, one-tap "year ago today" reshares.
- **Interaction depth:** more two-way formats (polls with results, Q&A chains), close-friends-first defaults to lower posting anxiety.
- **Consumption:** better tray ordering (favorites surface first), catch-up summary for infrequent openers.
- **Guardrail:** stories must keep driving DMs (the retention glue) - don't optimize views at the expense of replies.
- Metrics: % of DAU posting (creation rate), replies per story, story-completion rate, time-in-stories per session.

142. Improve Discord. Improvement

Discord's power (big open servers) is also its pain: noise, onboarding confusion, and moderation load.

- **New-user pain:** joining a big server is overwhelming - improve onboarding flows (choose interests - see only relevant channels first), server "start here" digest.
- **Signal vs noise:** better notification granularity (keyword alerts, thread-level follows), AI summaries of missed channels.
- **Moderator tools (the force multiplier):** better raid protection, AI-assisted flag queues, mod analytics - happy mods grow healthy servers.
- **Monetization balance:** expand Nitro value (server features, profiles) without paywalling community basics.
- Metrics: new-member week-1 message rate, server retention, mod action volume per incident, Nitro conversion.

143. Improve Twitch to make creators run more ads on the platform. Incentive design

Creators avoid ads because mid-roll ads drive viewers away; fix the trade-off, not the lecture.

- **Root cause:** ad breaks cost creators viewers (viewers leave during ads), so creators under-run ads even when paid.
- **Improvements:** picture-in-picture ads (stream stays visible - viewers don't leave), ad-break coordination tools (schedule at natural pauses, "ad break in 2 min" alerts), guaranteed revenue-share floors that de-risk experimentation.
- **Prove it:** show creators per-minute earnings WITH ads vs without using their own channel data.
- **Advertiser side:** better targeting and formats (sponsored segments, branded alerts) worth more per slot.
- Metrics: ad minutes per streamed hour, viewer retention through ad breaks, creator ad revenue, ad-fill rate.

144. Improve Gmail. Improvement

Email's modern pain is volume, not sending - triage and follow-through are the gaps.

- **Segment:** professionals drowning in newsletters, notifications, and CC threads.
- **Improvements:** stronger auto-triage (bundles you can kill with one unsubscribe), "follow-up needed" smart reminders (sent mail with no reply), AI summaries of long threads, better snooze tied to context ("remind me when I'm at work").
- **Trust guardrail:** any AI feature must be transparent (why is this in Important?) and reversible.
- **Quick wins vs bets:** unsubscribe-one-tap = quick win; agentic drafting = bigger bet.
- Metrics: time-to-inbox-zero, newsletter unsubscribe via Gmail, follow-up reminder action rate, NPS.

145. Make WhatsApp more user-friendly for the elderly. Improvement (segment)

Elderly users struggle with small UI, scams, and fear of "breaking something" - design for confidence.

- **Simple mode:** larger text/buttons, simplified home (just chats and calls), one-tap voice message (their preferred input), video-call shortcut per favorite contact.
- **Safety (critical for this segment):** scam warnings on unknown-number messages with links/money requests, "ask a family member first" friction on forwards, easy block/report.
- **Learning:** built-in first-run tutorial in regional languages, family "setup helper" flow (child configures parent's phone remotely with permission).
- **Voice-first:** voice typing, read-aloud messages for low-vision users.
- Metrics: 60+ DAU and calls per user, scam-report rate and blocked-scam rate, support requests, feature adoption of simple mode.

146. Improve Google Drive. Improvement

Drive's pain is finding things later - storage is cheap, retrieval is broken.

- **Segment:** heavy personal/team users with years of accumulated files.
- **Improvements:** natural-language search ("the deck Sam sent before the March launch"), auto-organization suggestions (project spaces instead of folder chaos), duplicate/version cleanup tools, storage-quota management with smart recommendations ("these 40 videos are 80% of your space").
- **Sharing hygiene:** expiring share links by default, a "who can see this" audit view - oversharing is a silent risk.
- **Quick win vs bet:** search understanding = high impact, hard; duplicate cleanup = quick, loved.
- Metrics: search success rate, time-to-file, cleanup-tool usage, share-audit adoption, storage upgrade conversion.

147. Improve TikTok. Improvement

TikTok owns discovery; its risks are creator economics, well-being pressure, and content depth.

- **Creator side:** more predictable monetization (beyond viral lottery), series/paid subscriptions for depth, better analytics.
- **Viewer well-being (also a regulatory shield):** smarter session limits, feed diversity controls, "why am I seeing this" transparency - pre-empts regulation and builds trust.
- **Utility expansion:** search (young users already search TikTok over Google - invest: better search results, Q&A formats), local services discovery.
- **Guardrail:** changes must not dent time-spent and DAU - test incrementally.
- Metrics: creator earnings distribution, search queries per DAU, session-limit opt-in retention, watch time (guardrail).

148. Improve Microsoft OneNote. Improvement

OneNote's strength is freeform capture; its weaknesses are sync reliability, search, and modern collaboration.

- **Fix the fundamentals first:** sync conflicts and version confusion erode trust - invest in reliable real-time sync before features.
- **Modern layer:** AI meeting-note summaries (ties to Teams recordings), templates for common workflows, better web clipping, handwriting-to-structured-text.
- **Collaboration:** comments and tasks on notes (today it's a solo tool in a collaborative world), smooth sharing to non-OneNote users.
- **Migration story:** importer from Notion/Evernote to capture switchers.
- Metrics: sync-error rate, weekly active notebooks per user, sharing rate, AI-summary usage, churn to competitors (exit surveys).

149. Improve Uber (overall). Improvement (broad)

Structure a broad question: pick the highest-leverage problem across rider, driver, and business.

- **Frame:** Uber's core tension - rider prices up, driver pay flat, Uber margin thin. Improvements should ease that triangle.
- **Rider:** price/ETA reliability (the top complaint: surge surprises) - pre-booked price guarantees, better pickup-point UX.
- **Driver:** earnings transparency upfront, smarter trip chaining (less dead mileage), deactivation-appeal fairness - driver retention = supply = lower ETAs for everyone.
- **Platform:** deeper membership (Uber One cross-sell rides+eats), high-margin scheduled rides and airport reliability.
- **Prioritize:** driver-supply health first - it compounds into every rider metric.
- Metrics: driver 90-day retention, ETA P90, rider repeat rate, Uber One attach, contribution margin per trip.

150. Improve Uber for a football event (match day). Improvement (event ops)

Match day is a predictable demand spike with a chaotic pickup zone - solve staging and communication.

- **Demand shaping:** pre-match "book your ride home" scheduling, post-match staggered-leaving incentives (discount for leaving 20 min later), group rides for fan groups.
- **Pickup-zone design (the big win):** designated Uber zones with lettered bays, in-app walking directions with AR/visual cues, PIN-matching (driver and rider confirm codes - no more "which white car?").
- **Supply:** guaranteed-earnings blocks for drivers serving the stadium window, staging-lot queue with fair ordering.
- **Safety:** verified-driver reminders, ride-share with friends tracking.
- Metrics: pickup time from request to car-moving, rider cancellation rate at venue, driver earnings/hour in zone, post-event NPS.

151. Improve Twitter (X). Improvement

Twitter's enduring pain: conversation quality and newcomer confusion vs its power-user density.

- **Conversation quality:** downrank rage-bait (optimize for "informed" signals, not pure engagement), better reply ranking, community-notes-style context at scale.
- **New-user value:** interest-graph onboarding (follow topics, get a good feed in day 1), lurker-first design (most users never tweet - serve reading).
- **Creator economics:** subscriptions and ad-share that make quality posting pay - retains the voices that make the network.
- **Trust & safety:** transparent, consistent moderation - advertiser and user trust are the revenue base.
- Metrics: time spent with satisfaction survey calibration, new-user D30 retention, creator payout volume, ad revenue per DAU with brand-safety incidents tracked.

152. Improve Spotify. Improvement

Spotify's strengths are personalization and ubiquity; gaps: artist economics perception, discovery beyond music, social.

- **Listening depth:** better playlist intelligence (auto-mixes per activity/mood), lyrics and context features that deepen engagement.
- **Social (underexploited):** real friend activity, shared blends, group sessions - music is social; Spotify is strangely solitary.
- **Creator side:** better artist-to-fan tools (direct merch/ticket links, fan analytics) - improves industry relations and content supply.
- **Podcasts/audiobooks integration:** cleaner separation so music UX stays pure while long-form grows.
- Metrics: listening hours per user, social-feature adoption, playlist save rate, podcast-to-music cross-consumption, churn.

153. Improve LinkedIn Jobs. Improvement

Job seekers' pains: ghost listings, black-hole applications, poor matching; recruiters' pain: unqualified volume.

- **For seekers:** freshness guarantees (remove stale listings), application-status transparency (viewed/rejected signals), fit-score with reasons ("you match 7 of 10 requirements"), salary ranges shown by default.
- **For recruiters:** skills-verified applicant ranking (assessments), spam-application filters, response-time SLAs surfaced to seekers (companies that ghost get labeled).
- **Network edge (unique to LinkedIn):** "2nd-degree connections at this company" with easy referral requests.
- **Trust guardrail:** fake-job detection - one scam does more damage than 100 good matches.
- Metrics: application-to-response rate, interview conversion per application, seeker NPS, recruiter qualified-candidate rate.

154. Improve Facebook for commuters. Improvement (context)

Commuting is dead time with flaky connectivity and one hand - design for that session shape.

- **Content packaging:** offline pre-load of feed (on WiFi at home), audio mode (listen to posts/videos), short-session "catch up in 5 minutes" digest.
- **Commuter utilities:** transit updates in the local tab, commute groups (carpool, delay alerts), saved-for-later queue that resurfaces at commute time.
- **Interaction:** one-hand friendly UI, voice comments, quick reactions over typing.
- **Data-light:** low-bandwidth mode default on metered connections.
- Metrics: commute-time session share, offline-mode usage, session completion (did they get through the digest?), return visits same-day.

155. Improve Google Photos. Improvement

Photos are piling up unorganized; the wins are curation, sharing friction, and rediscovery.

- **Curation:** AI auto-albums people actually keep (trips, people, pets), duplicate/blur cleanup wizard, "best of the month" auto-collections.
- **Sharing:** partner-shared libraries with granular controls (auto-share photos of the kids with spouse, not everything), live albums that update.
- **Rediscovery:** better memories with user control (hide dates/people), yearly recap, print/gift prompts at emotional moments.
- **Monetization guardrail:** storage upsell must stay honest - show clear management tools, not scare prompts.
- Metrics: album creation/keep rate, shares per user, memory-view rate, cleanup-tool storage freed, storage-plan conversion.

156. Improve Facebook tagging. Improvement

Tagging is a consent problem hiding inside a growth feature: tags spread content but can ambush the tagged person.

- **Consent:** default review-before-tag-appears (opt-out of pre-approval), easy bulk untag, tag-suggestion accuracy (mis-tags are the annoyance).
- **Value side:** face-recognition suggestions that are helpful (your own photos) with clear privacy controls; notify meaningfully ("you were tagged in 3 photos from Saturday").
- **Abuse:** tag-spam detection (marketing tags), harassment patterns (mass unwanted tags) rate-limited.
- **Balance:** tags drive distribution - measure the engagement cost of stricter consent before rolling out.
- Metrics: tag-approval rate, untag rate, tag-spam reports, distribution from tagged content (guardrail).

157. Increase the number of comments in Facebook groups. Growth (engagement)

Comments are conversation; more comments come from lower posting friction and faster first replies.

- **First-reply speed (the key loop):** notify likely responders, surface unanswered posts to active members - a post that gets a comment in 10 minutes gets ten more.
- **Lower friction:** comment prompts on posts (questions, polls auto-create comment threads), GIF/quick-reaction comments, anonymous-comment options in support groups.
- **Admin tools:** "conversation starter" post templates, weekly prompts, recognition of top commenters.
- **Quality guardrail:** comments must stay civil - toxic threads suppress future comments; watch report rate as you push volume.
- Metrics: comments per post, % of posts with 1+ comment in first hour, commenter retention, report rate (guardrail).

158. Improve Netflix. Improvement

Netflix's pains today: discovery fatigue ("nothing to watch" paradox), price sensitivity, and password-sharing friction.

- **Discovery:** mood/occasion-based browsing ("something light, 90 min, with partner"), human-curated shelves, better trailers-in-browse (playful but skippable).
- **Social light:** friends' recommendations (opt-in), watch-party revival - shared viewing drives retention.
- **Plans:** flexible tiers (mobile-only in price-sensitive markets), pause-instead-of-cancel, ad-tier experience quality (ad load honesty).
- **Content signals:** transparent renewal/cancellation info reduces "why start a doomed show" hesitancy.
- Metrics: time-to-first-play, browse abandonment, plan-downgrade vs cancel ratio, watch-party adoption, churn.

159. As PM of Amazon, reduce the rate of returns. Improvement (ops)

Returns are mostly expectation gaps (fit/quality/description) - prevent the mismatch before purchase.

- **Diagnose by category:** apparel (fit) vs electronics (compatibility) vs marketplace (misdescription) - different fixes per bucket.
- **Fit/size:** better size recommendation (purchase-history-based), "customers like you bought M" signals, true-to-size review summaries.
- **Listing quality:** enforce image/description standards for sellers, flag listings whose return-reason text matches "not as described", suppress known-bad SKUs.
- **Post-purchase:** setup guides for electronics (many returns are setup failures), easy exchange-instead-of-refund flow.
- **Guardrail:** don't make returns HARD - easy returns drive purchase confidence; reduce the need, not the option.
- Metrics: return rate by category, "not as described" share, exchange-over-refund ratio, category margin after return cost.

160. Flipkart earphones: high returns making the category unprofitable. What will you do?

RCA + category fix

A category economics problem - find whether it's product quality, expectation gap, or abuse, then act at the seller/listing level.

- **Diagnose:** return reasons coded (defective vs "not as expected" vs used-and-returned), by seller and price band - cheap unbranded sellers usually concentrate the problem.
- **Quality lever:** seller quality gates (minimum rating, sample testing for high-return SKUs), delist chronic SKUs, push certified/assured badging.
- **Expectation lever:** standardized spec cards (real battery hours, latency for gaming), Q&A and review highlights addressing common disappointments, compatibility checkers.
- **Abuse lever:** serial-returner detection, return-window and condition-check policies for high-fraud SKUs.
- Metrics: category return rate, return-cost per order, seller-level return distribution, category contribution margin, post-fix conversion (guardrail - don't kill sales).

161. Improve Facebook Marketplace. Improvement

Marketplace's pains: stale listings, flaky buyers/sellers, safety - trust infrastructure is the improvement theme.

- **Reliability:** auto-expire stale listings, response-rate badges on sellers, "likely sold" detection (see Q32), no-show feedback loop.
- **Trust:** ratings on completed transactions, verified profiles, safe meetup-location suggestions (police-station spots), in-app payment + shipping option with buyer protection for shipped goods.
- **Discovery:** better search (typo-tolerant, price filters), alerts for saved searches, local-deals feed quality.
- **Seller tools:** auto-description from photos, price suggestions, one-tap repost.
- Metrics: contact-to-transaction rate, listing freshness, no-show rate, scam reports per 10k transactions, repeat sellers.

162. Improve engagement on LinkedIn. Growth (engagement)

LinkedIn engagement is constrained by "professional anxiety" - people fear posting/commenting looks bad to their boss and network.

- **Lower participation anxiety:** comment prompts and safe formats (polls, "agree?"), anonymous-ish participation in groups, better "your post is performing" feedback to first-time posters.
- **Content quality in feed:** downrank engagement-bait ("comment YES"), elevate expertise signals - a better feed earns more return visits.
- **Habit loops:** notification relevance (only high-value), weekly digest of your industry's top posts, creator tools (newsletters, events) that give people a reason to post.
- **Segments:** job-seekers engage during search - give them career-content reasons to return between searches.
- Metrics: DAU/MAU, % of users interacting weekly, first-post conversion for lurkers, feed dwell time with satisfaction calibration.

163. Improve user engagement on WhatsApp. Growth (engagement)

WhatsApp is already a daily utility; engagement growth comes from richer conversation and new use cases, not more notifications.

- **Richer conversations:** polls, events in groups, better media (HD, albums), message editing/reactions - make groups more alive.
- **New surfaces:** Status improvements, Channels for creators/organizations (broadcast use case), communities for schools/apartments.
- **Utility engagement:** payments/business messaging where available - utility drives daily opens.
- **Guardrail (existential for WhatsApp):** privacy and no-spam - engagement tactics that feel like spam destroy the product's core trust; watch block/report rates.
- Metrics: messages per DAU, group participation rate, Status/channel adoption, business-message response rates, block/report rate (guardrail).

164. Improve Outlook for users returning from vacation to too many emails.

Improvement (specific pain)

The return-from-vacation inbox is a triage problem: 500 emails, 12 that matter.

- **Vacation digest (the headline feature):** on return, an AI summary - "what you missed" clustered by project/sender, with the 10 messages needing action highlighted; newsletters auto-bundled as one line.
- **During-vacation controls:** rules that hold non-urgent mail in a "later" folder, auto-decline meeting invites with delegation, OOO auto-replies that route senders to alternates.
- **Triage UX:** one-pass triage mode (archive/delegate/reply-needed decisions with keyboard speed).
- **Integration:** Teams/Calendar context (meetings you missed + recordings).
- Metrics: time-to-inbox-zero post-vacation (measure before/after), digest open and action rate, user stress-survey (CSAT for the feature).

165. Improve Amazon. Improvement (broad)

Broad prompt - pick the highest-leverage customer pain and go deep rather than listing everything.

- **Chosen pain: search result trust.** Sponsored results and review manipulation erode the "Amazon has what I need, reliably" promise.
- **Improvements:** clearer ad labeling and sponsored-organic separation, verified-purchase review weighting with fake-review detection, "Amazon's quality check" badge on tested products.
- **Supporting bets:** better comparison tools (side-by-side specs), delivery-date accuracy (a missed promise costs more than slow shipping).
- **Why this pain:** trust compounds - every other metric (conversion, Prime retention) sits on it.
- Metrics: search-to-purchase conversion, review-trust survey scores, return rate (proxy for expectation gap), Prime renewal.

166. Improve Facebook Birthdays. Improvement

Birthday wishes became low-effort noise ("HBD!") - the feature needs to help people be thoughtful again.

- **Quality over quantity:** prompt richer wishes - suggested memory/photo to attach, voice/video wish option, "schedule a call" nudge for close friends.
- **Notification design:** prioritize close friends' birthdays, remind earlier for gift-relevant relationships, charity-fundraiser birthday option (a proven mechanic).
- **Receiver experience:** a collected "birthday story" recap instead of 200 wall posts, easy one-tap thank-yous.
- **Measure both sides:** wish quality (responses per wish) and receiver delight.
- Metrics: % of wishes with media/personalization, reply rate to wishes, fundraiser creation rate, birthday-feature DAU.

167. Improve Facebook Stories. Improvement

FB Stories lag Instagram's; the opportunity is FB's different graph (family, older users, groups).

- **Lean into the FB graph:** family-moment prompts (photo memories, milestones), group stories (event/hobby groups), cross-post quality from Instagram without duplicate clutter.
- **Creation for non-creators:** templates and auto-created stories from photo libraries (one-tap share) - FB's users post less, so lower the bar.
- **Consumption:** tray ordering by closeness, mute controls, catch-up cards.
- **Guardrail:** don't cannibalize feed posting; stories should add sharing occasions.
- Metrics: creation rate among non-IG-linked users, story replies (interaction quality), time-in-stories, cross-post duplicate complaints.

168. Improve YouTube. Improvement (broad)

YouTube's standing pains: discovery beyond the algorithm bubble, creator burnout economics, ad experience tension.

- **Discovery:** more user control of recommendations (topic dials, "surprise me" mode), channel-following that actually means something (chronological subs feed option).
- **Creator health:** diversified revenue (memberships, shopping, courses), better burnout-reducing tools (scheduled publishing, AI editing) - creator retention protects supply.
- **Viewer experience:** honest ad-load management, better Premium value, comment-section quality (creator moderation AI).
- **Emerging:** living-room/TV experience and Shorts-to-long-form bridges.
- Metrics: watch time with satisfaction surveys, creator 12-month retention, Premium conversion, ad-opt-out rates.

169. Improve Google Maps. Improvement

Maps is utility-saturated; growth edges are richer real-world context and proactive assistance.

- **Proactive:** leave-now alerts with real traffic learning, trip chaining (errands optimized), parking prediction + payment.
- **Context richness:** better "vibe" data (crowdedness, queue length live), immersive/AR walking navigation, accessibility layer (step-free routes, wheelchair-friendly transit).
- **Local discovery:** trustworthy "best near me" (anti-review-fraud), personalized lists from friends.
- **India/emerging markets:** landmark-based directions, offline-first, two-wheeler mode depth.
- Metrics: navigation success rate, time saved vs baseline routes, discovery-tap-to-visit rate, accessibility-route usage, DAU.

Section 6 - Strategy & Business (Q170-Q195) - frameworks: value vs effort, build/buy/partner, unit economics, RICE

Pattern: strategy answers name the options, state decision criteria, commit to a recommendation with reasoning, and define what evidence would change your mind.

170. Airbnb PM: prioritize customers unable to check in vs customers with a dirty place.

Prioritization

Compare severity x frequency x fixability - and recognize one is trip-destroying, the other trip-degrading.

- **Impact:** unable to check in = the entire booking fails (refund, rebooking, stranded traveler, host conflict) - maximum severity per incident. Dirty place = bad experience but the stay continues.
- **Frequency & cost:** check-in failures are rarer but cost full rebooking + trust; cleanliness is more frequent and erodes reviews over time.
- **Decision:** prioritize check-in failure first (severity and immediate trust damage), with an emergency playbook (instant rebooking credit, 24/7 line), then attack cleanliness structurally (cleaning verification, review signals).
- **Show the matrix:** severity, frequency, user trust impact, cost to fix - interviewers want the reasoning visible.
- Metrics: check-in failure rate, rebooking time, cleanliness-complaint rate, post-incident rebooking retention.

171. How can retention be increased on a web/mobile job portal?

Retention strategy

Job portals have a built-in churn problem: success (getting a job) removes the user. Retention must come from career value beyond active search.

- **Fix search success first:** better match quality (skills-based recommendations), application-status transparency - users return to what works.
- **Habit hooks between jobs:** salary insights, skill-gap reports, market-demand alerts for their role, profile-view stats ("recruiters viewed you").
- **Alerts done right:** high-relevance job alerts at humane frequency - the #1 return driver and the #1 unsubscribe driver; tune relevance hard.
- **Long-term relationship:** career content, courses, resume tools - become the career home, not just the job board.
- Metrics: D30/D90 return rate of active seekers, alert open-to-apply rate, re-activation of placed candidates at 12-18 months (the next job change).

172. How can Lyft break into the private ambulance market?

Market entry

Clarify: this is non-emergency medical transport (NEMT), not 911 - a real, insurance-paid market that fits ride-hailing logistics.

- **Market fit:** NEMT = scheduled rides to dialysis, therapy, discharge; payers are insurers/health systems, not riders - Lyft's dispatch tech fits, but the buyer, reliability bar, and compliance (HIPAA) differ.
- **Entry strategy:** partner with NEMT brokers and health systems; dedicated trained driver pool (patient assistance certified); reliability SLAs far above consumer rides (missed dialysis = serious harm).
- **Build vs partner:** don't buy ambulances; extend the platform with healthcare credentials, scheduling, and billing integration.
- **Risks:** liability, regulatory licensing per state, brand risk if a patient is harmed.
- Metrics: on-time pickup (target 95%+), ride completion, cost per ride vs traditional NEMT, payer contract wins.

173. You own 6 pizza shops. Mobile app or Uber Eats to improve revenue? **Build vs buy**

At 6 shops, the own-app economics rarely beat marketplace reach - but delivery platforms tax every order.

- **Uber Eats pros:** instant demand and logistics, zero build cost; cons: 15-30% commission, you own no customer data, price competition on the shelf.
- **Own app pros:** margin kept, loyalty/data owned; cons: build+maintenance cost, you must generate your own demand and still need delivery logistics, app-install friction for a 6-shop brand.
- **Decision (state it):** start on aggregators for reach, simultaneously capture customer data (box inserts, loyalty program) and launch a light direct channel (web ordering or WhatsApp orders + third-party delivery) for repeat customers.
- **Numbers to compare:** commission per order vs app cost amortized per order; repeat-rate needed for the direct channel to win.
- Metrics: channel mix, margin per order by channel, repeat-customer share, loyalty signups.

174. How can you increase basket value for Uber Eats orders? **Growth levers**

Basket value grows through add-ons, bundles, and minimum-basket incentives - without hurting conversion.

- **Attach levers:** smart add-on suggestions at cart (drinks/dessert - highest margin), "complete your meal" prompts, combo building.
- **Threshold levers:** free delivery or discount above a basket threshold (moves users one item up), group ordering (multiple people, one order).
- **Discovery levers:** surface higher-value restaurants to users with spend history, premium categories (grocery, alcohol where legal).
- **Subscription:** membership (free delivery) raises order frequency; frequency x stable basket beats one-time basket bumps.
- Guardrails: conversion rate (thresholds too high = abandoned carts), delivery economics (bigger baskets shouldn't mean slower, worse deliveries). Metrics: AOV, attach rate, cart abandonment.

175. Compare Amazon Prime, Netflix and Hotstar. Favorite and why? **Comparison**

Compare on content strategy, business model, and moat - then commit to a pick with clear criteria.

- **Netflix:** pure subscription, global original-content machine, best personalization; weakness: single revenue line, content cost pressure.
- **Prime Video:** a perk inside the Prime bundle - retention weapon for commerce, not a standalone business; content quality inconsistent but improving; weakness: discovery UX.
- **Hotstar (India):** cricket/sports rights = unmatched live-audience gravity in India, plus regional content; weakness: churn between sports seasons, ad-tier experience.
- **Pick + criteria (example):** "Netflix as a product - recommendation quality and consistent content bar; Hotstar as a business in India - sports rights are the strongest moat." Any committed, reasoned pick scores.
- Structure tip: compare on the same axes, then pick, then state what would change your mind.

176. How would you monetize Wikipedia? Monetization with values

Wikipedia's trust comes from being free and ad-free - any monetization that breaks that kills the asset.

- **Constraint first:** no paywalling readers, no display ads that compromise neutrality (state this as the design constraint).
- **Options:** (1) donations done better - sustained memberships, employer matching; (2) enterprise API licensing - companies using Wikipedia data at scale (AI training, search engines) pay for guaranteed, structured access; (3) Wikimedia enterprise services (SLAs, support).
- **Also possible:** grants, endowment growth, merchandise/events - small but aligned.
- **Reject:** banner ads to readers - short-term revenue, long-term trust destruction; say why explicitly.
- Metrics: donor retention, enterprise contracts, revenue per reader without traffic/editor loss (guardrails: editor community health, donation sentiment).

177. How would you increase retention for Oyo Rooms? Retention strategy

Oyo's retention problem is trust: inconsistent quality means every booking is a gamble.

- **Fix the core first:** quality consistency - stricter property audits, verified photos, post-stay quality flags triggering delisting; retention follows reliability.
- **Loyalty mechanics:** membership with guaranteed standards + discounts (Wizard-style), corporate tie-ups for business travelers (the repeatable segment).
- **Use-case expansion:** long stays (students, remote workers), hourly stays, couple-friendly guarantees - reasons to return beyond occasional travel.
- **Service recovery:** instant rebooking/credit when a property fails - turning failures into saved customers.
- Metrics: repeat booking rate, NPS by property tier, % of stays meeting quality standard, membership enrollment and member repeat rate.

178. Upsell Uber Go to Uber Auto customers (PM at Uber/Ola). What product changes?

Upgrade funnel

Auto users choose price and availability; convert them at moments when Go's advantages are worth the delta.

- **Diagnose:** why Auto - price, availability in their area, habit? Upgrade only converts when the use case changes (rain, luggage, airport, with family, late night, AC in summer).
- **Trigger-based offers:** contextual prompts - raining = "Go at 20% off for this trip", airport rides, long-distance trips (Go comfort scales with distance).
- **Reduce first-trial risk:** first-Go-ride discount, price-lock comparison shown before booking ("Go only ₹40 more today").
- **Habit ladder:** after 2-3 Go rides in a month, offer a Go pass/subscription; measure cohort migration, not one-off conversions.
- Metrics: Auto-to-Go trial rate, repeat Go rate after trial, incremental revenue per user, guardrail: Auto supply shouldn't collapse (cannibalization).

179. Barber-service app wants to also deliver 20L water cans. New feature or new app?

Scope decision

Decide by user overlap, operational overlap, and brand - not by what's cheaper to build.

- **Criteria:** (1) Same user? Households booking barbers may also order water - plausible overlap. (2) Same operations? No - scheduled service appointments vs inventory logistics/delivery are entirely different stacks. (3) Brand stretch? A barber app selling water confuses positioning.
- **Options:** feature inside the app (fast, cheap, but diluted UX), new app (clean, but cold-start and marketing cost), or pivot decision (which business is actually better?).
- **Recommendation pattern:** validate demand cheaply first - a simple water-ordering section inside the existing app in one locality; if repeat orders prove out and ops diverge, spin into a dedicated experience.
- **Also ask:** does the core barber business have headroom? Don't diversify away from an unfinished main business.
- Metrics: cross-usage rate, water-order repeat rate, unit economics per delivery, barber-booking conversion (guardrail).

180. Amazon-Facebook partnership: use FB's friend network to drive Amazon sales. Walk through the lifecycle.

Partnership lifecycle

Treat a partnership like a product: discovery, definition, build, launch, measure, renew-or-exit.

- **Discovery/definition:** joint goal (social commerce), value split, data boundaries (who sees what - the hardest part: Amazon purchase data vs FB social data, privacy reviews).
- **Design the integration:** example - gift recommendations from friends' wishlists/interests, "friends also bought" signals, group gifting flows; consent UX is core, not polish.
- **Build & pilot:** API contracts, pilot in one market/category, legal/privacy sign-off.
- **Launch & measure:** attribution framework both sides trust (incremental sales, not just clicks), A/B against non-partnership baseline.
- **Renew or exit:** quarterly business reviews; exit criteria agreed upfront (data incidents, underperformance). Metrics: incremental GMV, gift-flow conversion, user consent rates, privacy complaints.

181. Outsourced critical mobile app: when do we take development in-house?

Build vs buy

In-house when the app is strategic and the hidden costs of outsourcing exceed the salary bill.

- **Signals to move in-house:** the app is core to competitive advantage; change requests outpace the vendor contract; knowledge lives only with the vendor (bus factor); quality/velocity suffering; total vendor spend approaching team cost.
- **Signals to stay outsourced:** requirements stable, app not differentiating, no budget to hire/retain a real team (a 2-person "in-house team" is worse than a good vendor).
- **Transition plan (asked or not, mention it):** knowledge-transfer period, codebase/documentation audit, hiring before terminating, parallel-run critical releases.
- **Decision rule:** in-house when (strategic importance x change velocity x vendor risk) exceeds (cost + hiring difficulty + management overhead).
- Metrics to justify: vendor cost trend, release lead time, defect rate, dependency incidents.

182. Should Uber Eats be a different app from Uber Rides? Portfolio strategy

Weigh cross-sell and shared infrastructure against focus and app simplicity.

- **For separate apps:** different use cases/moments (ride = urgent utility; food = browsing/leisure), cleaner UX per job, independent experimentation.
- **For one app:** cross-sell (ride users discover Eats = cheap acquisition), shared login/payment/data, one install, Uber One membership economics across both.
- **Decision criteria:** does bundling measurably lift cross-usage (check the data), or does it hurt conversion on each? Brand architecture and engineering cost of maintaining both paths.
- **Reasonable answer:** keep both apps but connect them (deep links, shared wallet, membership) - capture cross-sell without forcing one UX; say what evidence would flip you.
- **Metrics:** cross-app conversion rate, install cannibalization, order/ride frequency of dual users vs single-app users.

183. How would you increase revenue of MS PowerPoint? Revenue levers

PowerPoint monetizes through the Office bundle - revenue levers are attach, upgrade, and new audiences.

- **Upgrade levers:** premium AI features (Copilot design/drafting) behind higher tiers, premium template/stock marketplace with revenue share.
- **New audiences:** education and emerging-market tiers, individual creator subscriptions (today it's bundled for enterprises).
- **Retention levers (revenue = keeping the bundle):** collaboration depth (co-authoring parity with web rivals), designer features that make switching costly.
- **Adjacent revenue:** training/certification, enterprise template governance tools (brand compliance sells to big companies).
- **Metrics:** tier upgrade rate, Copilot attach, template marketplace GMV, churn of bundle subscribers.

184. LinkedIn feed: "Sponsored ads" vs "People you may know" on the top right?

Trade-off decision

Revenue now vs network growth later - decide with data and the company's stage, not ideology.

- **Frame:** ads = direct revenue per impression; PYMK = network density, which drives long-term engagement and retention (connections are LinkedIn's moat).
- **Decide by experiment:** A/B the slot; measure revenue per session AND downstream effects (connection rate, 30-day retention, session frequency). Often PYMK's long-term value exceeds the slot's ad CPM - but only measurement settles it.
- **Portfolio answer:** it doesn't have to be either/or - segment (new users see PYMK, they need connections; dense-network users see ads), rotate, or test hybrid units.
- **State assumptions:** if growth is the company priority, PYMK; if monetization pressure is high and network mature, ads.
- **Metrics:** revenue per session, connections added per session, retention by variant, ad CTR vs PYMK CTR.

185. Facebook Newsfeed: more ads vs "People you may know" widget?

Trade-off decision

Same structure as Q184 - applied to Facebook's mature network.

- **Key difference from LinkedIn:** FB's graph is mature/saturated, so marginal PYMK value is lower for most users; ad load tolerance is the real constraint.
- **Framework:** ad load has a user-experience cost curve - measure long-term retention impact per ad-load point in holdouts; PYMK adds friend-graph value mainly for newer/returning users.
- **Segmented answer:** PYMK for low-connection users (they need graph density), ads for dense-network users (highest tolerance and value), neither for ad-fatigued segments (protect retention).
- **Decision rule:** maximize lifetime value per user, not per impression - the metric that resolves this debate.
- Metrics: LTV by variant, ad-load tolerance thresholds, friend-acceptance rate, session-time changes.

186. Lyft wants to enter India. Strategy?

Market entry

India is a price-sensitive, auto-rickshaw-heavy, Ola/Uber-entrenched market - a straight copy of the US model fails.

- **Market realities:** low willingness to pay, cash/UPI payments, two-wheelers and autos as core supply, Ola/Uber entrenched, regulatory complexity per state.
- **Entry options:** partner/acquire (fast, expensive), niche entry (premium airport segment, women-safety-focused service, corporate contracts), or different vehicle category (bike-taxis where legal, autos).
- **Localization requirements:** UPI/cash, regional languages, low-end Android performance, driver economics tuned to Indian fuel/vehicle costs, safety features as a differentiator.
- **Recommendation shape:** enter 1-2 metros with a differentiated wedge (safety or corporate), prove unit economics, then expand - don't boil the ocean.
- Metrics: CAC vs LTV per city, driver earnings/hour vs incumbents, ride growth, take-rate sustainability.

187. How would you reduce cancellations for Uber?

Marketplace ops

Split rider cancellations and driver cancellations - they have different causes and fixes.

- **Driver cancels (usually the bigger issue):** causes - unprofitable trips, bad pickup locations, destination blindness. Fixes: upfront destination + fare transparency, cancellation penalties balanced with fair-pay floor, better pickup-point selection.
- **Rider cancels:** causes - long ETA, driver not moving, price regret. Fixes: honest ETAs, "driver en route" live progress, free-cancellation window design, re-match speed after a driver cancel.
- **Systemic:** matching improvements (don't assign far drivers), surge communication (sticker-shock cancels), cancellation-reason capture for data.
- **Incentive care:** harsh penalties push drivers off the platform - measure net supply effect, not just cancels.
- Metrics: driver/rider cancellation rate, post-cancel successful re-match rate, ETA accuracy, completed-trips impact.

188. Around 40% of Amazon reviews are fake. As PM, how do you tackle it? Trust & safety

Fake reviews are an adversarial market - you fight with detection, friction at the source, and incentives.

- **Detection:** ML on behavioral patterns (review velocity, account clusters, language similarity, reviewer-seller links), purchase-verification weighting.
- **Attack the supply:** sellers buying reviews - enforcement with teeth (delisting, fund holds), legal action against review-broker networks, monitor off-platform recruitment (Telegram/Facebook groups selling reviews).
- **Reduce impact:** rank verified, detailed, helpful-voted reviews higher; "vine"-style trusted-reviewer programs; review-summary AI trained only on verified reviews.
- **Incentive fix:** sellers fake reviews because ranking demands it - improve legitimate review-generation (post-purchase prompts) so honest sellers can compete.
- Metrics: % reviews removed/flagged, detection precision (don't nuke real reviews - guardrail), review-trust survey, seller enforcement counts.

189. PM at Uber: devise strategies to improve revenue. Revenue strategy

Structure revenue levers: more rides, more per ride, new lines, better take - each with an example.

- **Volume:** new segments (airport scheduled rides, Uber for Business, teens/seniors), geographic expansion, membership (Uber One) driving frequency.
- **Per-ride value:** premium tiers (Black, Comfort), dynamic pricing efficiency, in-ride add-ons (multi-stop pricing).
- **New revenue lines:** advertising (in-app, highly targeted by destination), delivery adjacencies (grocery, package), rentals/integrations.
- **Margin:** reduce incentives as marketplace matures, matching efficiency (less driver idle time = same supply, more rides).
- Prioritize by impact/effort with numbers (e.g. ads = high margin, medium build). Metrics: revenue per active user, take rate, ad revenue per DAU, Uber One member frequency uplift.

190. You are PM of Facebook Live. What are your priorities? Priority setting

Priorities depend on stage: prove the core loop (broadcasters go live - viewers watch - interaction - repeat) before monetizing.

- **Supply quality first:** broadcaster tools (stable streaming, easy go-live from any device, guest co-hosting), discovery so broadcasts find an audience (empty-room problem kills creators).
- **Viewer experience:** low-latency playback, live notifications for pages/people you care about, interactive formats (Q&A, polls, shopping).
- **Safety (non-negotiable for live):** real-time moderation - live video is the hardest content to police; invest before scale forces you.
- **Then monetization:** stars/tipping, brand deals, ad breaks once watch time justifies it.
- Metrics: broadcasts per day, concurrent viewers, interaction per viewer-minute, repeat broadcaster rate, violation detection latency.

191. Is a block in your neighborhood suitable for a new grocery store? How would you determine it? **Market analysis**

Classic site-selection: demand, competition, access, and economics.

- **Demand:** households and density within 1-2 km, income profile, demographics (families vs singles), current grocery spend leakage (where do they shop now, how far?).
- **Competition:** existing stores and their quality/distance, delivery-app penetration (the invisible competitor), underserved niches (fresh produce, ethnic foods, premium).
- **Access:** foot traffic, parking, visibility, delivery-logistics friendliness, rent vs projected revenue.
- **Economics:** estimate revenue (households x share-of-wallet x basket) vs costs; grocery margins are thin (1-3%) - volume and basket size decide survival.
- **Test cheaply:** survey residents, run a delivery-dark-store pilot from the location before committing to a full store.
- Decision output: go/no-go with the 2-3 decisive factors named.

192. How would you acquire more users for Uber? **Acquisition strategy**

Acquisition = new channels, new segments, new geographies, or converting adjacent users - pick by CAC vs LTV.

- **Referrals (best CAC):** two-sided referral bonuses tuned by city, timed after a great first ride.
- **Segments:** airport/travel partnerships, corporate accounts, seniors/teens programs, students (campus ambassadors).
- **Cross-sell:** Eats users to Rides and back - cheapest acquisition is an existing user.
- **Geography/supply-led:** launch in underserved cities where taxi pain is highest; ensure supply first (acquisition without cars burns money).
- **Discipline:** measure CAC payback per channel and kill channels that don't pay back in N months; beware promo-addicted users (retention by cohort).
- Metrics: CAC by channel, first-ride completion, D90 retention by acquisition cohort, LTV/CAC ratio.

193. Design surge pricing for Uber/Lyft. **Pricing design**

Surge is a marketplace balancing tool: it must clear the market without feeling predatory.

- **Mechanism:** real-time supply/demand per zone - multiplier rises when requests exceed available drivers; falls as drivers arrive. Communicate driver-side too (surge attracts supply - that's the point).
- **Design choices:** upfront pricing (show the final fare, not "2.3x"), caps for emergencies/disasters (ethics + regulation), smoothing to avoid whiplash changes.
- **Fairness/trust:** explainability ("prices are higher due to demand"), alternative offered (wait 15 min for lower price), predictability for scheduled rides.
- **Failure modes to name:** surge during emergencies (cap it), gaming by drivers (detect coordinated offline behavior), rider sticker shock driving churn.
- Metrics: request fulfillment rate, rider price-acceptance rate, driver supply response time, post-surge rider retention.

194. How do you say no to feature ideas or requests? PM craft

A good "no" protects focus and keeps the relationship - say no to the idea, not the person.

- **Anchor on strategy:** "Here's what we're optimizing for this quarter and why this doesn't move it" - a strategy makes the no objective, not personal.
- **Ask for the problem, not the feature:** often the request hides a real need solvable a cheaper way; dig before declining.
- **Transparent criteria:** show the prioritization framework (RICE/value-effort) and where their idea lands; offer the backlog as a visible parking lot.
- **Close the loop:** tell them the decision and reasoning, invite them back with data; a no without feedback creates shadow work.
- Also know when to say yes cheaply: small asks that buy goodwill with key partners are sometimes worth it - say this too; blanket no's are as bad as yes's.

195. How do you decide between building a feature or fixing a bug (same effort)?

Prioritization

Compare user impact, not category labels - a bug affecting many beats a feature few will use.

- **Quantify both:** bug = users affected x severity x frequency (revenue/trust lost now); feature = expected adoption x value per user (future gain).
- **Weight trust:** bugs in core flows compound (churn, support cost, reputation); features rarely compound in reverse.
- **Consider timing:** contractual commitments, launch windows, bug's trend (growing vs stable).
- **Rule of thumb:** core-flow bugs affecting revenue/trust first; cosmetic or edge bugs can wait behind a high-value feature. State the exception logic explicitly.
- Framework on paper: impact x reach x confidence for the feature vs severity x affected users for the bug - same units (users impacted x how much) makes them comparable.

Section 7 - Technical (Q196-Q198) - explain clearly, then name the design trade-offs

PM technical questions test whether you can reason with engineers, not whether you can code. Give the big picture first, then the components and trade-offs.

196. What happens when you enter a URL in your browser? Systems explainer

Walk the chain: lookup, connect, request, render.

- **1. DNS lookup:** the browser resolves the domain to an IP address (cache - OS - resolver - authoritative DNS).
- **2. Connection:** TCP handshake, then TLS handshake (encryption, certificate check) - HTTPS.
- **3. Request/response:** browser sends an HTTP GET; the server (often via CDN/load balancer) returns HTML.
- **4. Rendering:** browser parses HTML into the DOM, fetches CSS/JS/images (more requests), builds the render tree, paints pixels; JavaScript may fetch more data (APIs) and update the page.
- **PM-relevant extras:** caching and CDNs (why speed varies), status codes (404/500), redirects, cookies/auth, and where latency hides (DNS, TLS, server time, heavy JS) - useful when discussing performance with engineers.

197. How does TinyURL work? System design

A URL shortener is a lookup service: store long-URL by short key, redirect fast.

- **Core:** generate a short unique key (counter converted to base62, or hash of the URL + collision handling), store key - long URL in a database, and respond to short-link hits with an HTTP 301/302 redirect.
- **Scale reasoning:** reads vastly outnumber writes - cache hot links (Redis), CDN edge redirects; database is simple key-value.
- **Capacity (Fermi moment):** 7-character base62 keys give ~3.5 trillion combinations - plenty; say why 6-7 chars suffice.
- **Extras:** custom aliases, expiry, analytics (click counts - dedupe bots), abuse scanning (short links hide phishing - safety matters).
- Design trade-offs to name: hash vs counter (collisions vs predictability), 301 vs 302 (caching vs analytics accuracy).

198. Create an API design for third-party payment integration. API design

Payments APIs live or die on reliability and idempotency - money must never move twice or get lost.

- **Core endpoints:** create payment intent (amount, currency, customer, idempotency key), confirm/capture, refund, status query, webhook for async events (success/failure).
- **Idempotency:** every charge request carries a unique key; retries with the same key never double-charge - the single most important design point.
- **Security:** API keys + signed webhooks, tokenized card data (never handle raw cards - PCI scope), TLS everywhere, least-privilege scopes per key.
- **Reliability & DX:** clear error codes (insufficient funds vs fraud vs system), sandbox environment, rate limits, versioning (v1, sunset policy), great docs and SDKs - developer experience is the product.
- Metrics as PM: integration time to first successful transaction, success rate, dispute handling, developer NPS.

Section 8 - PM Craft & Behavioral (Q199-Q204) - answer honestly, with structure and a real example

Behavioral answers score on authenticity + structure. Use a real story where possible (STAR: situation, task, action, result).

199. How do you know when to cut corners to get a product out the door?

Judgment

Cut scope deliberately, never silently - and never on trust, safety, or money.

- **Never cut:** security, privacy, payments correctness, data loss risks - these debts come back with interest.
- **OK to cut:** polish, secondary features, edge-case coverage (with logging), internal-tooling elegance - things you can iterate on after learning.
- **The test:** is this corner reversible, and does shipping now teach us something worth the debt? If the corner is invisible to users and recoverable, take it; if users will feel it in a core flow, don't.
- **Make it explicit:** log the debt, get team/stakeholder agreement, schedule the payback - undocumented corner-cutting is how products rot.
- Have a real example ready (a time you shipped an MVP with a manual backend step, then automated it after validation).

200. How would you keep developers motivated and turning out quality work?

Leadership

Engineers are motivated by mastery, autonomy, purpose, and not-wasting-their-time - a PM influences all four.

- **Purpose:** connect work to users - share customer stories and metric impact; nothing motivates like "your fix helped 40k users".
- **Respect their time:** clear requirements (fewer mid-sprint surprises), defend focus time, kill meetings that could be docs.
- **Autonomy & craft:** involve engineers in the "why" and early solutioning (not just tickets), support tech-debt time, celebrate quality (testing, reliability wins) not just launches.
- **Fair credit:** name engineers in wins, take blame for unclear specs yourself.
- Watch for burnout signals (velocity drops, quality slips, quiet disengagement) and address workload, not morale-posters.

201. In layman's terms, describe your day-to-day as a Product Manager.

Behavioral

A PM decides what gets built and why, then clears the path for the team building it.

- **Morning:** check the numbers (dashboards, overnight issues) and clear blockers for the team.
- **Core work:** talk to users (calls, feedback, data) - decide priorities - write requirements for the next thing - align with design and engineering on trade-offs.
- **Glue work:** sync with stakeholders (sales, support, leadership) so what ships matches what the business needs and promises.
- **The honest summary:** a PM owns outcomes, not output - you're accountable for whether the thing worked, and you do it through influence, not authority.
- Keep it concrete: one real day from your experience beats the generic list.

202. What is your favorite product? Why? Behavioral (product sense)

They're testing product thinking, not taste - pick something you genuinely use and analyze it like a PM.

- **Structure:** what it does - the user problem it nails - the specific design decision that makes it great - what you'd improve (shows critical thinking, not fanboying).
- **Example skeleton:** "Google Maps - turns anxiety about the unknown into confidence; the killer detail is live traffic re-routing, which quietly saves millions of hours daily; I'd improve offline-first behavior for emerging markets."
- **Pick something with texture:** a product you can discuss in detail beats a trendy pick you can't defend. Lesser-known picks are memorable if the analysis is sharp.
- **Avoid:** praising without a user problem, or listing features without the "why it matters".
- Prepare 2-3 (one mainstream, one niche, one you dislike-and-why) - interviewers sometimes ask follow-ups.

203. A metric for a video streaming service dropped by 80%. What do you do?

Root-cause funnel

An 80% collapse is almost certainly instrumentation or a catastrophic bug - check measurement before anything else.

- **Step 1:** which metric exactly, and when did it start? An instant cliff = deploy or logging change; a fast slope = real behavior.
- **Step 2:** cross-validate with independent data (CDN bandwidth, server play requests, revenue) - if those are flat, the metric is broken, not the business.
- **Step 3 (if real):** segment hard - platform, region, content; playback failure (DRM/license outage, regional CDN failure, crash on play) is the likely culprit at this magnitude.
- **Step 4:** incident-manage: roll back or fail over, communicate, postmortem with monitoring gaps closed.
- Rule of thumb: the bigger the drop, the more likely it's measurement or infrastructure, not user preference. (Similar pattern: Q40, Q52.)

204. How would you improve Google Maps? Improvement

Maps is utility-saturated; the edges are proactive assistance and richer real-world context.

- **Proactive:** leave-now alerts with real traffic learning, trip chaining (multi-errand optimization), parking prediction + payment.
- **Context richness:** live crowdedness and queue lengths, AR walking navigation, accessibility layer (step-free routes, wheelchair-friendly transit).
- **Local discovery:** trustworthy "best near me" with anti-review-fraud, friend-curated lists.
- **Emerging markets:** landmark-based directions, offline-first, two-wheeler mode depth.
- Metrics: navigation success rate, time saved vs baseline, discovery-to-visit rate, DAU. (Same question family: Q134-Q169 improvement pattern.)