PM Interview study notes - Xào

Table of Content

Overview

This document summarizes my approach to the PM interview at tech companies like Google and Amazon. While I have used all sorts of resources out there and have tons more personal notes, only the most important information is included here. I suggest you go through the materials listed and prepare your own frameworks / notes.

My 3-phase approach

This summarizes the approach if you have about **7 months to prepare**. If you have less time than that, you can shorten the amount of time needed for each phase.

Phase 1: from 0 to 1

Goal: understand and remember key knowledge required to perform well in a PM interview
Timeline: 2.5 months, 10-20 hours / week
Focus: Product Design & Technical
Note: *Before the start of phase 1, please read through 2 books (1) Cracking the PM interview (2) Decode and conquer book.*

Material used:

- Cracking the PM interview - reference only. Don't remember framework from this book
- Decode and conquer: simpler, straightforward frameworks to use as a start
- The design of everyday things: skim through, this book was recommended by multiple PMs but not something that would be too useful for PM interviews. Read if curious.
- Grokking the system design interview course on educative.io: for technical interviews only. Do not need if you are not interviewing for Google PM or Technical PM at other companies

- Stratechery blog: great blog on tech trends and strategy. I always use this to start forming an opinion on tech trends / news. Useful for strategy type of questions. Write down notes as you read good articles related to the company you are interviewing for!

Weekly plan summary

Week	Topics to research & prepare own notes / framework
1	**[Product Strategy]** [1] What's the company's goals? [2] What's the market opportunity? Porter Five Forces [3] What's the success metrics? [4] How to do competitor analysis? [5] What's a good product? [6] Prioritizing products
2	**[Technical] Data** [1] Data structures (Arrays, Linked List, Trees, Graphs, Hash table, Stacks, Queues). [2] Database designs: pros and cons. -- Indexing -- Replication -- Tranactions -- Keys, triggers, join types -- Design decision
3	**[Product Design] Steps to designing product features** [1] User segmentation / persona [2] Customer Journey and needs [3] Customer research methodology (UXR) [4] Analysis of features / solutions - 10x solutions. [5] Design - High-level UX flow, instrumentation of analytics events → Create own framework based on existing CIRCLEs framework
4	**[Product strategy] Steps to take product to market** [1] Phases of launches [2] Determine success metrics [3] Optional - Marketing strategy & Community strategy [4] Others - security, privacy, technical scalability, down time, internationalization, legal, etc. **[Analytical] Metrics and Data collection** [1] Discovery > Activation > Engagement > Monetization > Ads [2] Collect data: product usage data, surveys, qualitative 1:1 or group feedback, A/B testing **[Analytical] Metrics dips and spikes - how to analyze issues and trends** [1] Follow up questions on the data & Root problem analysis [2] How to communicate with stakeholders? [3] How to come up with remediation steps for teams to tackle?
5	**[Technical] Operating systems** [1] Processes, threads, concurrency issues.

	[2] Locks, mutexes, semaphores and monitors how they work. [3] Deadlock and livelock and how to avoid them [4] What resources a process needs, and a thread needs, and how context switching works, and how it's initiated by the operating system and underlying hardware. [5] Fundamental of modern web design
6	**[Product design] How to think 10x - bringing it together** [1] IDEO methodology [2] Trends in different industries (lateral innovation) **[Technical] System design** [1] Features sets, interfaces, class hierarchies [2] Designing a system under certain constraints, simplicity and robustness, trade offs [3] API design
7	**Web knowledge** [1] Cookies, Sessions [2] HTML, CSS, XML, JSON [3] REST vs. RPC [4] DNS, TCP/IP [5] HTTP / HTTPs [6] Web vulnerabilities (XSS, SQL injection) [7] AJAX [8] UDP
8	Review interview frameworks and questions on Metrics and data collection Review Opearting systems questions
9	Review interview frameworks and questions on How to think 10x Review System design questions
10	Review Web knowledge questions Review all types of product questions

Phase 2: from 1 to 2

Goal: finalize all frameworks for PM design, strategy, analytical, and technical interviews; Practice in details 1-2 questions / day and schedule for mock interviews in phase 3.

Timeline: 2 months, 10-20 hours / week

Focus: Everything

Material used:

- All of your notes / framework from Phase 1. Note: as you practice more questions in phase 2, your frameworks will continue to change. In fact, my framework keeps changing until the last few weeks of my mock interview.
- **Exponent** course (https://www.exponent.com/): highly recommend to go through all of their videos on Youtube as well as in the course. Some are great, some are so-so. The ones with Facebook PM are great! I even downloaded it and listened to that one multiple times to get used to the style and how short / long my answer should be.
- **https://www.productmanagementexercises.com/** (some answers are not so great here, but a few popular ones are good, and list of questions are good)

- **The product manager interview** book by Lewis Lin: I did not read this entire book, only skim through to get a list of interesting questions.
- **Technical book**: I honed into technical details in phase 2. Two books that are great:
  - Swipe to unlock: fun analogies to help you explain tech in layman terms
  - Designing Data-Intensive Applications: I skipped the last few chapters
- **Grokking the system design interview course** on educative.io. If you can understand everything in here, you are all good for the system design part of the technical interview.

Weekly plan summary

Week	Topics
1+2	**Framework to review & finalize:** Product design (based on CIRCLES but modified as I practice) **Sample questions** - Design a smart shoe - Design a high-tech gym - Design camera for the blind - Design a teleportation machine interface - Design the fridge for the future
3	**Framework to review & finalize:** Product improvement **Sample questions** - Pick a product and tell me how you would improve it? - Improve Google Maps / Google Photos / Gmail / Youtube - Improve garage door opener / dishwasher
4	**Framework to review & finalize:** [Analytical] Estimation **Sample questions** - How many police officers are there in the US? - How much storage do you need for Google Maps - What's the potential market size for [product X]?
5+6	**Framework to review & finalize:** [Analytical] Metrics & diagnosing metrics problem **Sample questions** - Youtube's video engagement dropped by 15%. What would you do? - How would you measure success of [Google Calendar / Youtube / etc.]
7+8	**Framework to review & finalize:** [Strategy] Tradeoff, new market entry, CEO-level issues and strategy, pricing **Sample questions** - Should Google enter the grocery industry / ride-sharing market / etc.? - What should LEGO do in 1 year? 5 years? 10 years? - What is an area that [Google / or any other company] underinvested? - Should Youtube enter the [Africa / Portugal / etc.] market? - Why did Google acquire Fitbit?

Phase 3: Home run

Goal: do lots of mock interviews, refine framework after each practice
Timeline: 2.5 months, average 2-3 mock interviews / week
Focus: Everything

***Note**: I suggest doing mock interviews with calibrated PMs (those who have done interviews before and know how to evaluate candidates). Peer-interviews can be useful too so you can see different approaches people take; and they are often better when you are not confident yet to practice with PMs. However, in the last 1-2 months before your interview, mock with calibrated PMs! They can greatly improve your performance.*

Material used:

- Own notes from previous phases
- You will probably don't need books or other materials at this stage. Mostly just practice, practice, practice, and refine your own frameworks.

Plan summary

- **Self-practice:** spend the **first two, three weeks** of phase 3 just practicing on your own to be fully ready for mocks. Each day, practice 3-4 questions and write down answers in detail.
- **Mock interview:** The following number of mock interviews are recommended for Google PM interview, which is one of the toughest out there. For PM interviews in other companies, you might need less.
  - At least 3-5 technical mock interviews if you are interviewing for PM positions that require technical knowledge
  - At least 5-15 product design mock interviews
  - At least 5 strategy mock interviews
  - At least 5 analytical mock interviews
  - At least 5 mix-of-everything interviews

Simplified frameworks

My detailed frameworks are quite long since I add all the notes / suggestions I have from my mock interviews. I suggest you build up your own framework this way too: start with simplified frameworks / frameworks you read from books, then modify and add more notes as you practice.

DESIGN

1. **Clarify**: market, timing, constraints
2. **Assume goals** & why does it matter for Google
3. ***[Optional if technology is given] Clarify technology**: Size, cost, who can use it, limitation (pros & cons)*
4. ***[Optional if product is vague] Value chain + Competitor**: to decide where to play.*
5. **Target users & choose one.** List at least 3 groups of users / persona
6. **User Journey & painpoints +** rank
   a. Go beyond functional needs (e.g. emotional needs)
7. **Solution brainstorming +** rank
   a. Have at least 1 moonshot idea.
   b. Go through basic ideas quickly → interviewers probably already heard it countless times
   c. Paying attention to user trends and current events is a plus
8. ***[Optional if monetization question] Pricing model:** fremium, bundling, pay-as-you-go, subscription, razor and blades, tier-*
9. ***[Optional if strategy question] Strategy recommendation:** for what to build in short vs. long-term*
10. **Evaluation & metrics**
    a. Concerns
    b. How to measure success of your chosen option
11. **Summary & Elevator pitch**: Be good at storytelling here! Create an exciting bigger mission.

Notes: a less common type of question if "free brainstorming", meaning you are given a technology, and just have to list out as many ideas to commercialize that technology as possible, within 10-15 minutes. There is no framework for this, just practice them and write down your own approach on which areas you like to think about for these types of open-ended questions (e.g. which industries, which problems, etc.)

STRATEGY - It's hard to have one framework for strategy type of questions. Below is a general list of frameworks I will pick and choose depending on question types.

1. **Clarify** product, timeframe
2. **Goals**
3. **Frameworks used (choose 1 or more depending on questions)**
   a. 3Cs
   b. Porter five forces
   c. Value chain analysis
   d. PEST
   e. ANSOFF matrix
4. **Evaluate different strategy**: what's the tradeoff
5. **Strategy recommendation**
   a. Suggest both short + long term strategy

Important note: One frequently asked strategy question is "Why is product X important for company Y?" You won't find any framework for this question in any of the books, so I suggest reading and practicing questions to come up with your own criteria to evaluate why a specific product is important strategically for a company.

ANALYTICAL

Measure success

1. Clarify products & main actions
2. Clarify users
3. Company objectives
4. Metrics
   a. Basic AARRR framework
   b. Or split by business vs. user-focused framework
   c. Tips: to stand out, come up with **unique metrics** outside of those basic AARRR metrics.
5. [Brownie point] Concerns about the metrics / when does the metrics fail

Debug changes in metrics

1. Clarify definition
2. Understand context & go through potential problems
   a. External factor: competition, industry, environment, users behavior, etc.
   b. Internal factor: which products / which features? Reporting issues, etc.
3. Hypothesis & how to test for those hypothesis
4. Suggest what to do to solve the situation

Estimation questions

1. Clarify each vague word of the prompt
2. [Optional] Highlight various approaches to solving the problem (top down, bottom up) → chose 1
3. Write down detailed formula, remember to split by different user groups if applicable
4. Calculate

    a. Tips: keep a separate list of assumptions for easy tracking

5. Gut check:
    a. Does the answer seem right? If not which assumptions could be wrong?

FACT SHEET (for estimation question)

Populations

World population: ~7.5B
Asia Population: 4.5B
China population: ~1.4B = India population
US population: ~330M
California population: ~40M
San Francisco, CA Population: ~900K
Los Angeles, CA Population: ~4M
NYC population: ~8.5M
Seattle, WA population: 750K

U.S. Specific Facts

Size of continental US: 3M square miles (1 square mile = 26 square km)
# of US households: 100M
Average people per household in US: 3
Life expectancy: ~80 years
Median household income: ~$60K

Product User Populations

Internet: US: 90% - World 60%
Smartphone penetration: US: 70% - World: 50%
Android market share: US: 40% (vs. 60% iOS) - World: 70% (vs. 30% iOS)
Fraction of Americans that own a smartwatch/fitness tracker: 1/5 people
Consumers vs. commercial vehicle on highway: 1:20
1.5 cars / household
Number of global Amazon Prime Subscribers: ~150M
Instagram monthly active users: ~1B (2018)

Technical

Video size per minute: 2MB (480p) - 5MB (720p)
Image size: 1-3MB → take 2MB
Average Click Through Rate (CTR) for a search ad: ~2%
Amazon AWS S3 (object storage) Standard cost: $0.023 / GB / month → $0.25 / year
Average landing page conversion rate: 2%
Average WiFi bandwidth: ~12-25Mbps

Measurement

1 ounce ~ 2 tablespoon
1 cup = 8 ounce. Rice: 250 gram / cup

Time

9000 hours / year (25*350)
500K minutes / year

TECHNICAL - System design only

1. Requirements: functional & non-functional
2. General consideration: call out anything important, top of your mind about the system (e.g. read:write ratio, traffic, etc.)
3. High-level system design
   a. Explain data schema, general architecture choice, and big blocks (clients, app servers, database, and how they talk to each other)
4. Scaling elements:
   a. Add elements that help the system scale (caching, Load balancers, sharding, etc.)

Tips

1. **Where do I find a list of questions?**
   You can find a lot of sample PM questions (but not necessarily answers) from:
   - **Exponent** course
   - **https://www.productmanagementexercises.com/**
   - **The product manager interview** book by Lewis Lin

   In total I suggest practicing 70-100 questions on your own, 2-3 times each (in details), plus probably 50+ more from mock interviews. If you can get this much practice, by the time you are in the actual interviews, pretty much all questions are somewhat similar to other questions you have practiced before (though PM interviewers are very creative so they will surprise you with new types of questions all the time! So please practice to be better at handling multiple types of questions, NOT to memorize solutions)

2. **How to brainstorm new ideas quickly, especially moonshot ideas?**
   Keep a list of your favoriate ideas! Build up this list by reading a lot of new product ideas on pages like ProductHunt, and by researching new technology and startup ideas. Not as part of the PM practice, but as an unrelated side project (on xaoytuong.com), I read & summarized probably close to 500 startup ideas, as well as practiced using frameworks like SCAMPERs, Whati If, and Oblique strategies to come up with modified products from existing products. This turned out to be super helpful for me when it comes to practicing for design questions. While I can't share this list, it's not hard to build one for yourself (or get a box from my website - Xào ;))
   In the last week leading to the design interview, I suggest reviewing your favorite idea list everyday.

3. **What I wished I practiced earlier**
   [i] Brainstorm user segmentation: having meaningful user segments is not easy. Anyone can list out mutliple users, but whether they are interesting and meaningful or not needs practice.

   Additionally, there are multiple ways to split users: geographically, demographically, behaviorally, etc. Some people also prefer creating a user persona rather than splitting users by one dimension. Choosing which way to segment users, thus, can be tricky depending on the situation. Practicing more will help!

   [ii] Brainstorm user needs: Seems simple at first, your success in design questions can really depend on the user needs you identify. If it's interesting, obviously painful, but not obvious at first, then it will help open a good discussion for solutions and help you stand out. If you highlight things that everyone else thinks about, then you pretty much have to have outstanding ideas to be memorable.

I suggest going through all sorts of users you can think of, as well as different ways to segment users. Then for each segment, brainstorm unique pain points. This exercise will help you in multiple questions when you see the same user segment you have thought about comes up.

4. How do I stand out?

Have your own frameworks: modify from existing framework. Add something interesting / different. Everyone has read these common books so you can't rely on just that to do well.

Be confident: being posed, confident, and calm during interviews helps a lot. Smile if you can :) If you don't know how to answer and get stuck → prepare for these cases with some sentences you can use to buy you time to think (asking for a minute to think is GOOD). Do not rush into talking or answering a question unless you have an idea on where you are going.

Think about the users first, always: PMs have to be user-centric. Any trade offs being discussed need to focus on the user first, then others later.

Think about the big picture: The higher level you are aiming for, the more strategic thinking is required. So when doing tradeoffs between ideas / options, always discuss the general strategy and how an idea is good for the company & the users.

Always evaluate: your ideas / calculation / strategy will always have loopholes. A good PMs will always think about these drawbacks / loopholes without prompting from interviewers.

Have good moonshot ideas: you are also evaluated based on your creativity. If you are normally not a creative person, practice it! (See Tips #2 above) Also note that moonshot does NOT equal AI / ML / IoT → everyone would think about these. Think of moonshots as something not obvious + a blend of new technology is a plus.